

TADM10_2

SAP NetWeaver AS Implementation & Operation I

SAP NetWeaver

Date _____
Training Center _____
Instructors _____

Education Website _____

Participant Handbook

Course Version: 62
Course Duration: 10 Day(s)
Material Number: 50089098



An SAP course - use it to learn, reference it for work

Copyright

Copyright © 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Trademarks

- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.
- ORACLE® is a registered trademark of ORACLE Corporation.
- INFORMIX®-OnLine for SAP and INFORMIX® Dynamic Server™ are registered trademarks of Informix Software Incorporated.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.
- Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® is a registered trademark of Sun Microsystems, Inc.
- JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- SAP, SAP Logo, R/2, RIVA, R/3, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Disclaimer

THESE MATERIALS ARE PROVIDED BY SAP ON AN "AS IS" BASIS, AND SAP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THESE MATERIALS AND THE SERVICE, INFORMATION, TEXT, GRAPHICS, LINKS, OR ANY OTHER MATERIALS AND PRODUCTS CONTAINED HEREIN. IN NO EVENT SHALL SAP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND WHATSOEVER, INCLUDING WITHOUT LIMITATION LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS OR INCLUDED SOFTWARE COMPONENTS.

About This Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.




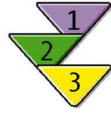

Typographic Conventions

American English is the standard used in this handbook. The following typographic conventions are also used.

Type Style	Description
<i>Example text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths, and options. Also used for cross-references to other documentation both internal (in this documentation) and external (in other locations, such as SAPNet).
Example text	Emphasized words or phrases in body text, titles of graphics, and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, and passages of the source text of a program.
Example text	Exact user entry. These are words and characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Icons in Body Text

The following icons are used in this handbook.

Icon	Meaning
	For more information, tips, or background
	Note or further explanation of previous point
	Exception or caution
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Course Overview	viii
Course Goals	viii
Course Objectives	viii
Unit 1: Technology Components for Browser-Based User Dialogs	1
Internet Scenarios with SAP Systems	3
Appendix: SAP Internet Transaction Server (standalone)	9
Internet Communication Manager	22
Internet Communication Framework	40
The SAP Web Dispatcher	71
Load Balancing in the SAP NetWeaver AS Java Environment	98
Unit 2: Basics of User Administration AS ABAP	113
User Administration Concept	115
Authorization Concept	126
Login Parameters and User Info	141
Appendix: Advanced User Administration Topics	151
Unit 3: User and Authorization Concept AS Java	161
Architecture and Configuration of the User Management Engine (UME)	163
User and Group Administration	187
The Java Authorization Concept	203
Special Principles	216
Unit 4: RFC Connections	231
Fundamentals and Variants for Using RFC	232
Setting Up RFC Connections	237
Unit 5: Communication and Integration Technologies	253
Cross-System Business Processes	255
Remote Function Calls and BAPIs	260
Enterprise Services-Oriented Architecture (Enterprise SOA)	270
Web Services	276
SAP Business Workflow	280

Unit 6: Working with SAP Solution Manager.....	293
Concept of the SAP Solution Manager	294
Connecting ABAP-Based Systems to SAP Solution Manager.....	305
Unit 7: System Monitoring and Troubleshooting AS ABAP	353
Monitoring Architecture	355
Including Remote Systems	369
Creating Your Own Monitors	377
Properties Variants and Threshold Values	386
Trace Options	398
Troubleshooting Procedure.....	410
Unit 8: Monitoring AS Java	419
Java Monitoring: Overview	421
Monitoring SAP NetWeaver AS Java	429
Appendix: Background Information About the Monitoring Service.....	447
Connecting to a Central Monitoring System	458
Log Viewer and Log Configuration	473
Availability Monitoring	505
Appendix: Statistics and the Performance Trace.....	519
Appendix: Solution Manager Diagnostics (SMD).....	541
Glossary	563
Index.....	569

Course Overview

Course TADM10 is the foundation for various, further training courses for consultants. After TADM10, you can continue your training to become a (Technical) XI or EP Consultant. Alternatively, you can proceed to course TADM12 where you will further expand your knowledge of SAP NetWeaver AS.

This training content is **largely independent of the type of operating system and database technology**.

To pass the exam to become a certified **SAP Certified Technology Associate - System Administration - SAP NetWeaver 7.0 (<database>)**, you must have good knowledge of the content of courses TADM10 and TADM12, as well as one of the following database-specific courses: TADM51 (Oracle), TADM53 (MS SQL Server), TADM56 (DB2 on Win/UX) or ADM515 (MaxDB).

Like the other TADM courses, TADM10 comprises several individual courses (or parts thereof), which are arranged here in a way that will enable you to gain the knowledge you require as an SAP Technology Consultant as efficiently as possible.

Week 1 of course TADM10 is based on content taken from the following courses:

1. SAPTEC - Fundamentals of SAP NetWeaver Application Server
2. ADM100 - Administration AS ABAP I
3. ADM200 - Administration AS Java

Week 2 of course TADM10 is based on content taken from the following courses:

1. SAPTEC - Fundamentals of SAP NetWeaver Application Server
2. ADM100 - Administration AS ABAP I
3. ADM102 - Administration AS ABAP II
4. ADM200 - Administration AS Java
5. And some new course material, which is currently only covered in week 2 of TADM10.

At the end of the database-specific part of this training (TADM5#, or after ADM515, to be booked separately), there is a three-hour certification exam that covers topics from courses TADM10 and TADM12 and TADM5#/ADM515.



Caution: Note that the certification exam has been designed in such a way that the answers to all of the exam questions are contained in the folders provided for courses TADM10, TADM12, TADM5# (or ADM515).

Therefore, you do not require any additional course material even if the instructor hands out other books during the course or provides additional information not contained in the course folder.

Target Audience

This course is intended for the following audiences:

- SAP Technology Consultants (Associate Level)

Course Prerequisites

Required Knowledge

- Basic knowledge of IT
- Basic knowledge of operating systems and databases



Course Goals

This course will prepare you to:

- To work as a Technology Consultant
- To configure and manage SAP Web AS ABAP
- To configure and manage SAP Web AS Java



Course Objectives

After completing this course, you will be able to:

- To process administrative tasks in SAP systems

Unit 1

Technology Components for Browser-Based User Dialogs

Unit Overview

In this unit, you learn about a number of central technology components that are important if SAP systems are used for intranet or Internet applications. This course focuses on managing the components introduced here; development of these components is covered in other courses. The lesson entitled “Load Balancing in the SAP NetWeaver AS Java Environment” discusses some topics already covered in the previous lessons. Therefore, the main focus of this lesson is client-based load balancing.

- The SAP Internet Transaction Server (ITS) is used with Web applications (IACs) and with SAP GUI for HTML. Depending on the system release and scenario in question, the functions of the SAP ITS can be implemented by means of a standalone ITS or using the ITS integrated in the AS ABAP.
- The Internet Communication Manager (ICM) is the process that turns the conventional ABAP application server into a Web server or Web Client.
- The Internet Communication Framework (ICF) provides an environment for handling HTTP(S) requests in the ABAP work process using Web applications such as BSPs.
- With the usage type AS Java, the SAP NetWeaver Application Server provides a complete runtime environment for J2EE applications.
- SAP Web dispatcher distributes HTTP(S) requests to a suitable application server (instance).



Unit Objectives

After completing this unit, you will be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the areas of use of SAP ITS, ICM, AS ABAP, and AS Java
- Describe the architecture of the SAP ITS (standalone)

- Perform simple administrative tasks on an SAP ITS
- Describe the implementation area of the ICM
- Configure and monitor the ICM
- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Activate and use the integrated ITS as of AS ABAP 6.40
- Outline the function of the SAP Web Dispatcher
- Explain how you can use the SAP Web Dispatcher to distribute workload across the different instances of an SAP system
- Explain how load balancing can be realized in the SAP system

Unit Contents

Lesson: Internet Scenarios with SAP Systems	3
Lesson: Appendix: SAP Internet Transaction Server (standalone)	9
Lesson: Internet Communication Manager.....	22
Exercise 1: Administration of the ICM	31
Lesson: Internet Communication Framework.....	40
Exercise 2: Administrative Work with the ICF	57
Lesson: The SAP Web Dispatcher	71
Exercise 3: Administration of the SAP Web Dispatcher.....	81
Lesson: Load Balancing in the SAP NetWeaver AS Java Environment.....	98

Lesson: Internet Scenarios with SAP Systems

Lesson Overview

SAP provides a number of ways in which applications can be created for intranet or Internet users. This lesson introduces the technologies on which these applications are based and explains the differences between them.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the areas of use of SAP ITS, ICM, AS ABAP, and AS Java

Business Example

Your company wants to allow its customers browser-based access to data in the SAP system (for example, in the context of Web-based purchasing). As a member of the system administration team, it is your task to compare and evaluate different methods of realizing this.

SAP Internet Transaction Server (SAP ITS)



User Access

- SAP GUI
- Web browser and mobile devices through SAP ITS

User Interface

- Screen

Programming Language

- ABAP

Communication Interface

- RFC
- Third-party products through connectors and gateways

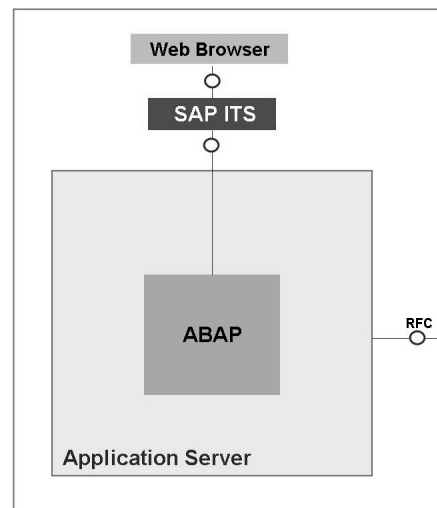


Figure 1: As of SAP Basis 3.1G: Web-Enabling Using SAP ITS

SAP delivered the first version of the SAP Internet Transaction Server (SAP ITS) with SAP R/3 3.1G in 1996. It is a software component that acts as a gateway between a Web server and an SAP system. SAP ITS switches between Internet protocols and formats (such as HTTP, HTTPS, and HTML) and those of the SAP system (such as DIAG, RFC, and screens).

First, the SAP ITS was implemented as standalone software, that was used “before” an ABAP-based SAP system. This “standalone” ITS existed as of Release 3.1G up to and including 6.20 (upwardly and downwardly compatible with SAP systems up to and including SAP Web AS 6.40). As of SAP Web AS 6.40, the new ITS is integrated in AS ABAP on all platforms with a simplified architecture.

Web applications that were developed specifically for SAP ITS are called Internet Application Components (IACs). These include Employee Self Services (ESS) that are based on SAP R/3 and SAP R/3 Enterprise or the SAP Online Store. The SAP GUI for HTML also uses the SAP ITS.

SAP ITS is therefore required for **existing Web applications (in IAC technology) and the SAP GUI for HTML**, regardless of the basis release of the corresponding SAP system.

Internet Communication Manager (ICM)



User Access

- SAP GUI
- Web browser and mobile devices

User Interface

- Screen
- BSP (Business Server Pages)

Programming Language

- ABAP

Communication Interface

- RFC
- HTTP(S)
- SMTP
- SOAP/XML

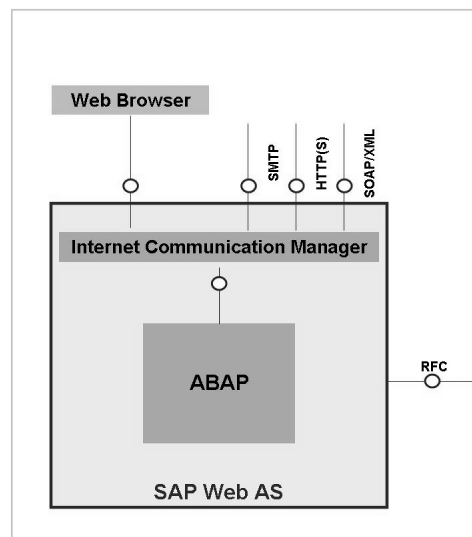


Figure 2: As of SAP Web AS 6.10: Openness Using the ICM

Based on the highly-scalable infrastructure, new technologies are used as of SAP Web AS 6.10 to process HTTP requests (and other protocols) directly from the Internet or to send HTTP client requests to the Internet. To achieve this, the SAP Kernel has been extended with the Internet Communication Manager (ICM) process.

The ICM process forwards requests to the Internet Communication Framework (ICF), which supports numerous programming models. This is how the SAP CRM, SAP BW, and SAP XI software components use this infrastructure. A programming model for such applications are the Business Server Pages (BSPs).

AS Java



User Access

- SAP GUI
- Web browser and mobile devices

User Interface

- Screen
- BSP (Business Server Pages)
- JSP (Java Server Pages)

Programming Language

- ABAP
- Java

Communication Interface

- RFC
- HTTP(S)
- SMTP
- SOAP/XML

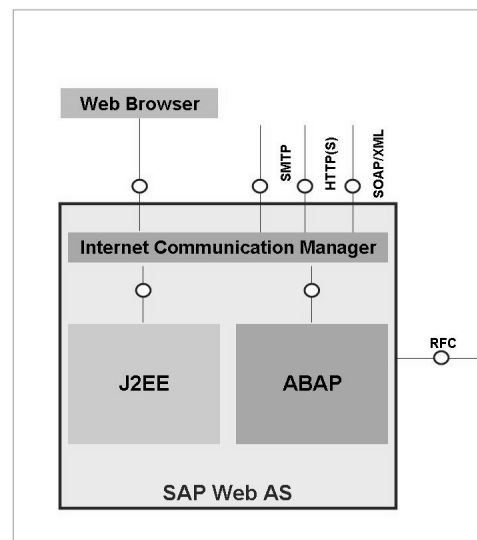


Figure 3: As of SAP Web AS 6.20: Integrated Java Runtime Environment

With the AS Java, SAP has a complete J2EE-compatible application server in its product range. The SAP NetWeaver Application Server provides the following installation options (as of Release 6.20):

- SAP NetWeaver Application Server ABAP (AS ABAP)
- SAP NetWeaver Application Server Java (AS Java)
- SAP NetWeaver Application Server ABAP+Java (AS ABAP+Java)

Developers, therefore, have a mature development and runtime environment for **applications based on Java**. Examples of SAP software components that use the J2EE engine include SAP NetWeaver Portal (usage type SAP EP), SAP Exchange Infrastructure (usage type SAP PI), and custom functions in SAP Customer Relationship Management (SAP CRM).

Note that the J2EE standard not only describes the (browser-based) user dialog, but also specifies a complete application server.

Web Dynpro



User Access

- SAP GUI
- Web browser and mobile devices

User Interface

- Screen
- Web Dynpro for Java
- BSP (Business Server Pages)
- JSP (Java Server Pages)

Programming Language

- ABAP
- Java

Communication Interface

- RFC
- HTTP(S)
- SMTP
- SOAP/XML

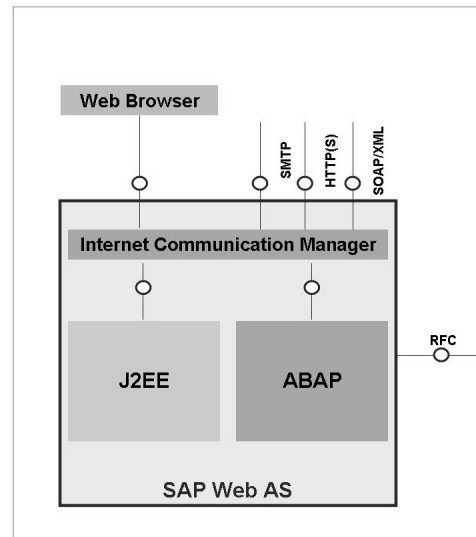


Figure 4: As of SAP Web AS 6.40: Web Dynpro for Java



User Access

- SAP GUI
- Web browser and mobile devices

User Interface

- Screen
- Web Dynpro ABAP
- Web Dynpro Java
- BSP (Business Server Pages)
- JSP (Java Server Pages)

Programming Language

- ABAP
- Java

Communication Interface

- RFC
- HTTP(S)
- SMTP
- SOAP/XML

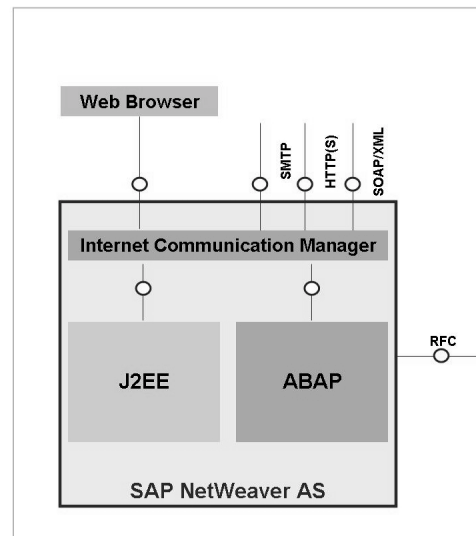


Figure 5: As of SAP Web AS 7.00: Web Dynpro ABAP

Web Dynpro is the preferred programming model for business application Web interfaces in SAP systems based on SAP NetWeaver. It provides a clear distinction between the user interface (UI) and the business logic. It also provides functions that are not usually available as part of the standard tools for developing professional user interfaces. These include functions for checking entries, providing input help, supporting multiple languages, and handling errors comfortably, as well as caching mechanisms that ensure fast response times and are therefore especially useful for interactive user interfaces.

The Web Dynpro programming model is available in the Web Dynpro instances for Java (as of AS Java 6.40) and Web Dynpro for ABAP (as of AS ABAP 7.00). The basic concepts of these two instances are very similar, and so the user cannot recognize the technology used.

For more information about the Web Dynpro, see the SAP Developer Network at <https://www.sdn.sap.com/irj/sdn/webdynpro>.



Lesson Summary

You should now be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the areas of use of SAP ITS, ICM, AS ABAP, and AS Java

Related Information

- Quick Link `/ui` (for User Interface) on the SAP Service Marketplace

Lesson: Appendix: SAP Internet Transaction Server (standalone)

Lesson Overview

After a short introduction to the architecture of the SAP ITS (standalone), this lesson will show you the various options for administration. This section is classified as an appendix in the current version of the ADM102 course and is relevant only for those customers who still use a standalone SAP ITS (due to the product releases in operation).



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the architecture of the SAP ITS (standalone)
- Perform simple administrative tasks on an SAP ITS

Business Example

Your company is making the browser-based SAP GUI for HTML available to some of its employees. As a member of the system administration team, it is your task to ensure the availability of the SAP Internet Transaction Server, through which requests from the SAP GUI for HTML access the SAP system. Since the connected SAP systems are based on SAP Web AS 6.20 and earlier, the conventional “standalone” SAP ITS 6.20 is used.

Architecture of the SAP ITS (Standalone)

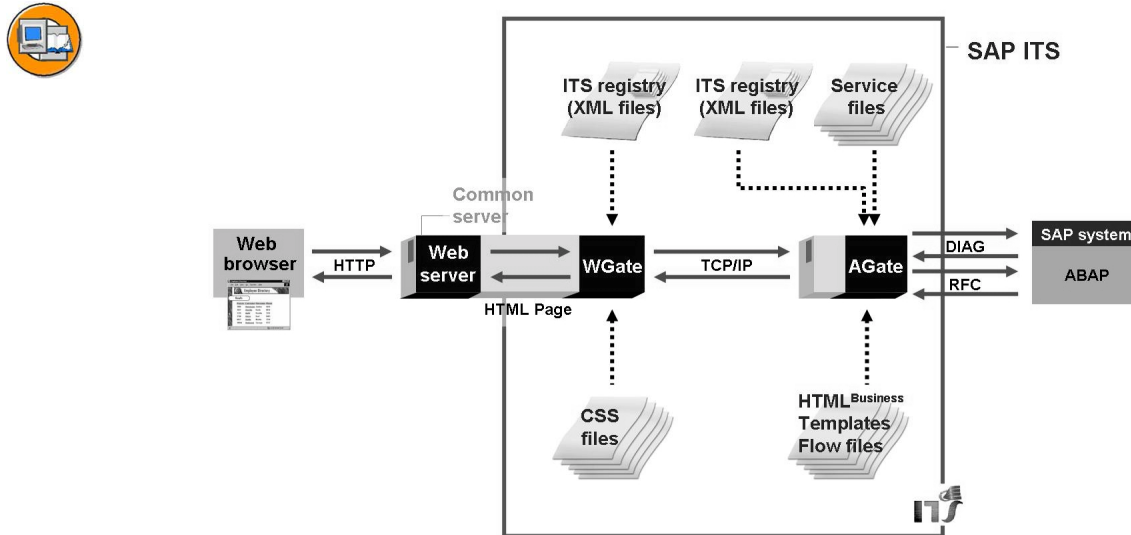



Figure 6: Architecture of the SAP ITS (Standalone)

A Web server (from third-party vendors) is required to operate the SAP ITS standalone. The SAP ITS standalone itself consists of two components, the **WGate** (Web Gateway) and the **AGate** (Application Gateway).

An HTTP(S) request is processed in the following steps:

1. The request is sent from the user's Web browser to the Web server using the HTTP(S) protocol.
2. The Web server recognizes that the request is for the SAP ITS from the structure of the requested URL. The request is sent to the WGate (by Apache module, NSAPI protocol, or ISAPI protocol depending on the Web server provider), which is implemented as a filter in the Web server.
3. The WGate transfers the request to an assigned AGate.
4. Service files (simple ASCII files) on the AGate determine which function is started in which component system.
5. The requested transaction or the requested function module is executed in the SAP system.
6. The AGate converts the output into HTML, either using templates (called HTML Business Templates) or dynamically at runtime, as is the case with SAP GUI for HTML.
7. The formatted data is sent to the user's Web browser through the WGate and the Web server. Formatting information, such as font, font size, or colors, can be stored separately in cascading style sheet (CSS) files. If appropriate, additional MIME objects (such as images, audio, or video files) are then downloaded by a Web server.

In SAP ITS 6.20, the **ITS Registry** (*Registry.xml* file with other subfiles in the ITS *config* directory) is the storage location for the configuration settings.

 **Note:** In SAP ITS 6.10 (which is no longer supported), the WGate is configured using a *wgate.conf* file and the AGate is configured using Windows Registry entries. The *wgate.conf* file still exists in 6.20, but links to the ITS registry.

The SAP ITS standalone is **very scalable**, which allows many installation options. As a basic principle, the Web server and WGate always run on the same host. The WGate and AGate are available for the following platforms (October 2007):

WGate		AGate
Operating System	Web Server	Operating System
Windows 2000 Server Editions (SP 1 or higher)	Microsoft Internet Information Server 5.0; Sun ONE Web Server Version 6.0 and 6.1; Apache HTTP Server: Versions 1.3.14, 1.3.19, 1.3.26, 1.3.27, and 2.0.XX	Windows 2000 Server Editions (SP 1 or higher)
Windows 2003 Server Editions	Microsoft Internet Information Server 6.0; Sun ONE Web Server Version 6.0 and 6.1; Apache HTTP Server: Versions 1.3.14, 1.3.19, 1.3.26, 1.3.27, and 2.0.XX	Windows 2003 Server Editions
SuSE Linux Enterprise Server 8 and 9 (Intel)	Apache HTTP Server: Versions 1.3.14, 1.3.19, 1.3.26, 1.3.27, and 2.0.XX	SuSE Linux Enterprise Server 8 and 9 (Intel)
RED HAT Enterprise Linux 3 (Intel)		RED HAT Enterprise Linux 3 (Intel)
SUN Solaris 8.0, 9.0, and 10.0	Sun ONE Web Server Version 6.0 and 6.1	SUN Solaris 8.0, 9.0, and 10.0

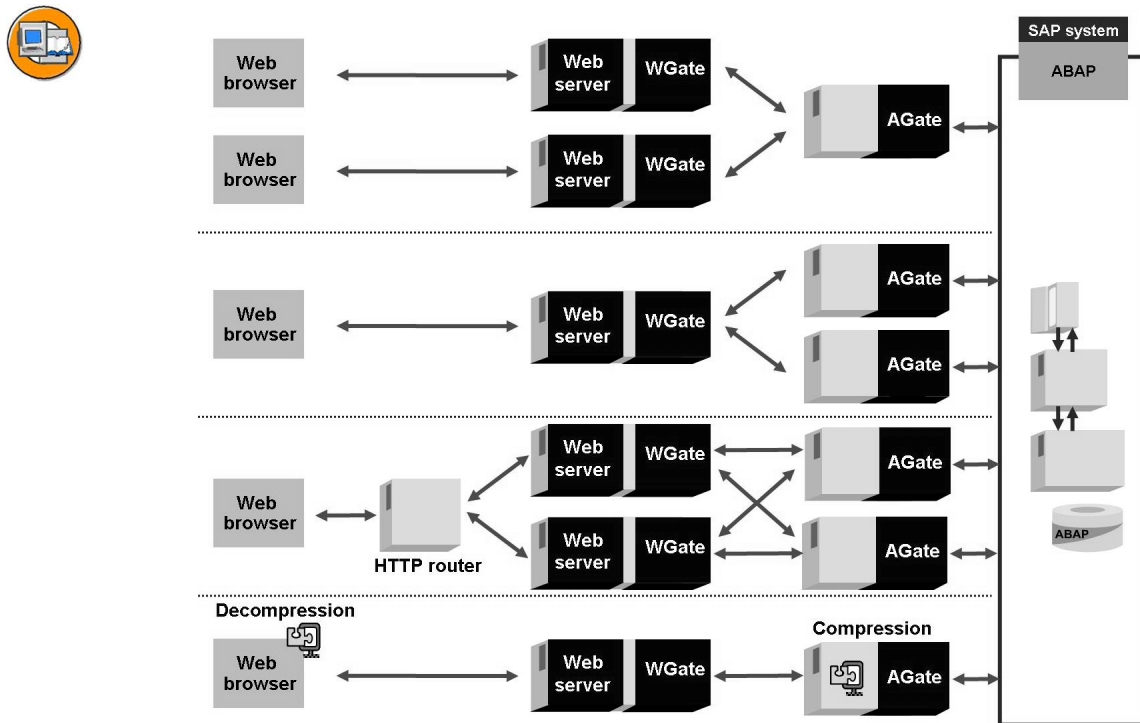


Figure 7: Scalability of the SAP ITS Standalone

- External products (HTTP routers, Web switches) can distribute incoming requests over multiple Web servers.
- The WGate and AGate can run together on one host (single host) or on separate hosts (dual host).
- You can run multiple (virtual) SAP ITS instances on a single host.
- A WGate can control multiple AGates.
- An AGate can be addressed by multiple W Gates.
- You can set the dimensions (number of workthreads and memory size) of each AGate for the expected user load.
- An AGate can log on to the SAP system using load distribution (logon groups).
- The AGate can reduce the volume of data transferred to the Web browser (compression with *gzip*).

We strongly recommend that you work with a current patch level of SAP ITS 6.20; earlier versions of SAP ITS are not supported. The SAP ITS is backward compatible, meaning that, for example, you can use SAP ITS 6.20 with an SAP R/3 4.6C installation without any problems. For more information about the maintenance strategy, see SAP Note 197746.



Note: At the time of writing (October 2007), SAP Note 197746 specifies the (earliest) end of maintenance for SAP ITS 6.20 as March 31, 2013.

WGate Administration

As stated, an SAP ITS cannot be run without a Web server. SAP supports various products on a range of platforms. For information about administration of the Web server, contact your product's vendor.

For the WGate, the ITS registry (*Registry.xml* file in the ITS *config* directory) refers to subfiles for configuring the IAC Object Receiver (IACOR) (see below) and the WGate. The *ITSRegistryWGATE.xml* file therefore contains information such as the list of ITS instances, the AGate server for executing these instances, and their parameter settings. The WGate is thus able to forward requests to the relevant AGates.

You can change the configuration by opening the ITS registry with a text editor of your choice and changing the settings to suit your requirements. For configuration changes to take effect, you must then restart the Web server.

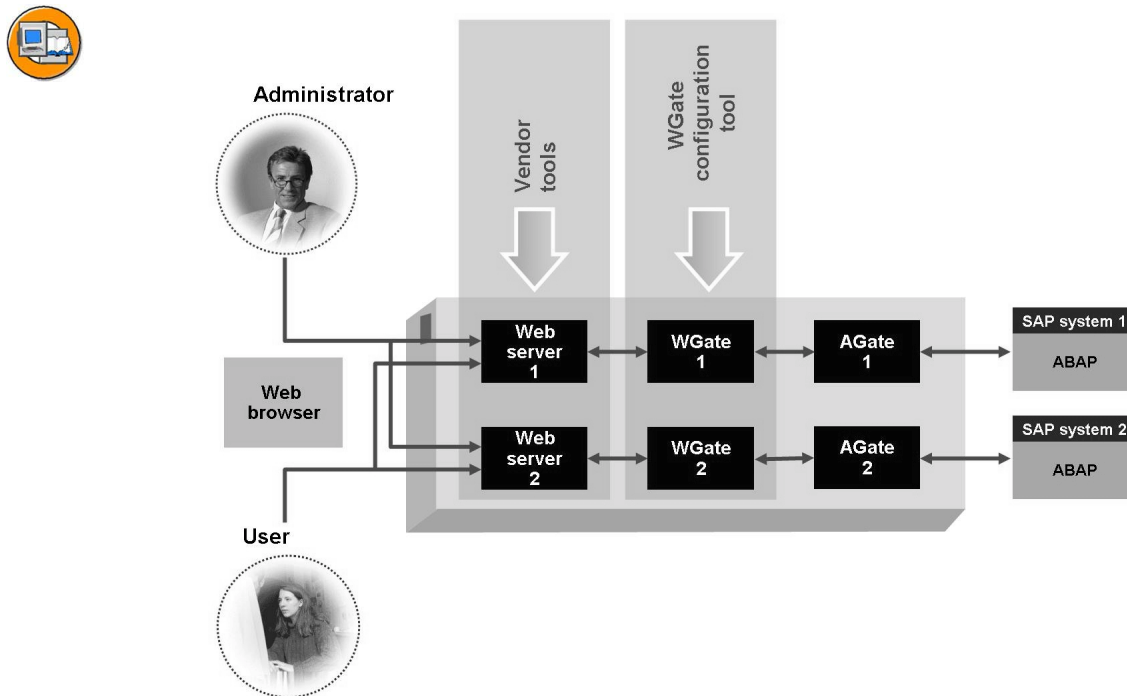



Figure 8: Administration of the Web Server and WGate

The browser-based **WGate configuration tool** is significantly easier to use. Administrators can use it to display and change the ITS registry without having to understand the exact structure of the XML files and the actual parameter names. Proceed as follows to use the WGate configuration tool:

1. Open the *ITSRegistryWGATE.xml* file with a text editor of your choice.
2. Set the *ConfigMonitorEnabled* attribute from **no** to **yes**.
3. Enter the URL **http://<server with domain>:<SID ITS port>/scripts/wgate/wgate-restart**.
 ➔ **Note:** The *ITSRegistryWGATE.xml* configuration file is then reloaded (only if changes have been made). The Web server is not restarted.
4. You can now start the WGate configuration tool by entering the URL **http://<server with domain>:<SID ITS port>/scripts/wgate/wgate-config**.
5. Once you have changed and tested the settings, set *ConfigMonitorEnabled* to **no** again to avoid incorrect configuration by other users.
6. Enter the URL **http://<server with domain>:<SID ITS port>/scripts/wgate/wgate-restart** again.

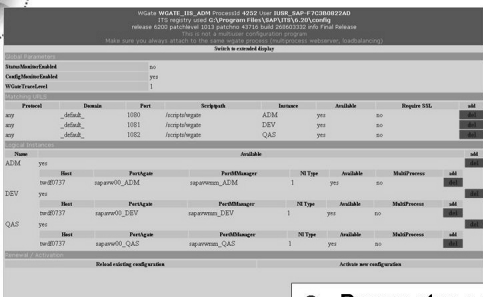
For more information about the possible parameters and their significance, see the online documentation. For more information about the WGate configuration tool, see SAP Note 688295.





Admin

<http://<server with domain>:<ITS port>/scripts/wgate/wgate-config>



- Parameter settings
- Matching URLs
- AGate connection data
- AGate availability test

Figure 9: WGate Configuration Tool Functions

AGate Administration

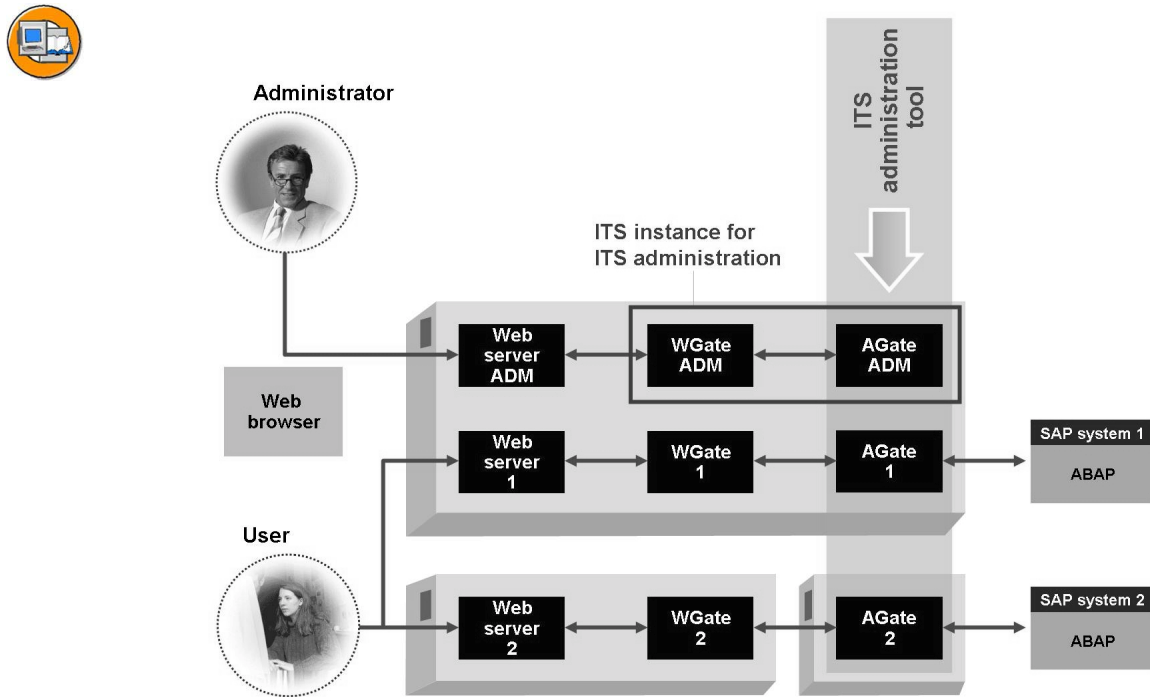


Figure 10: Administration of the AGate

The **ITS administration tool** is also browser-based. It allows the administration of SAP ITS AGate instances. During the installation, you can create a separate SAP ITS instance for the ITS administration tool (recommended approach). You can call tools using a special URL (requires at least one operational SAP ITS instance – see the figure above). The *itsadmin* user is an initial user, whose password is set by default to *SAPinst* at the time of the installation. This user is characterized by the fact that it is the only user that can create additional users and assign authorizations (for example, restrict a user to display functions only).

Since SAP ITS 6.10, you can also monitor AGates of a virtual SAP ITS on remote hosts, although there are functional restrictions when compared to local SAP ITS instances.

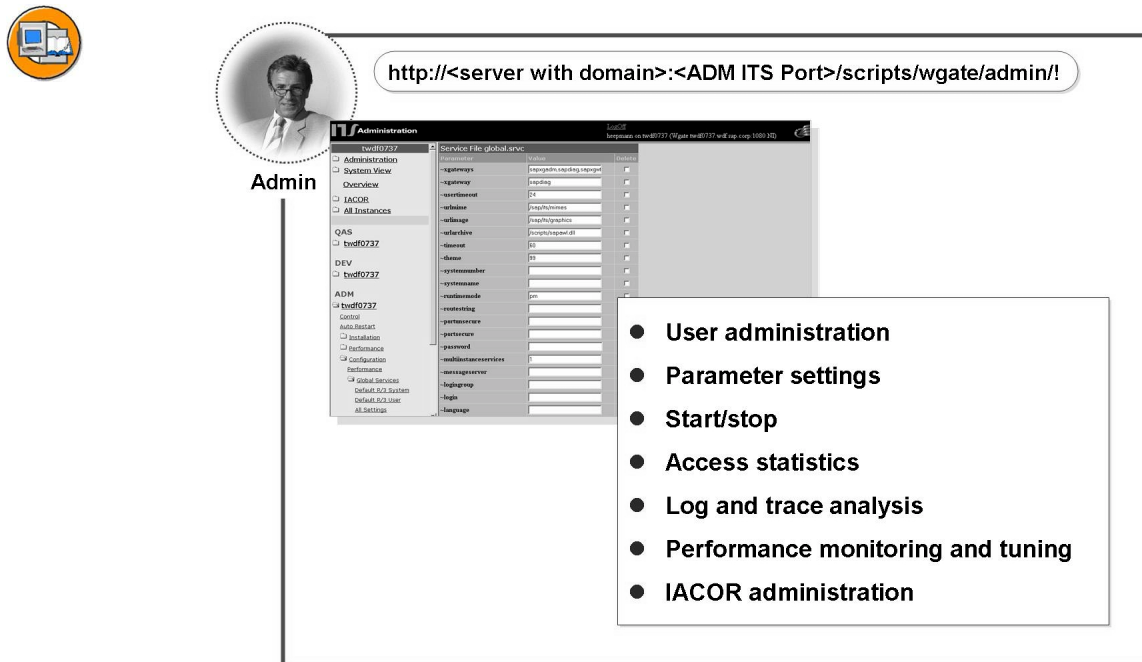


Figure 11: Functions of the ITS Administration Tool

The following list contains a number of selected functions of the ITS Administration tool:

- User administration
- Configuring all SAP ITS parameters
- Starting and stopping the AGate and IACOR
- Evaluating various log and trace files
- Performance monitoring and tuning

Outlook: Other Aspects in the SAP ITS Standalone Environment

There are various additional components and themes surrounding the SAP ITS standalone, which will be mentioned here only in brief.

IAC Development

Two tools are available for developers to create and edit IAC objects:

Inside an SAP system, the **Web Application Builder for ITS Services** allows IAC objects for ITS services to be edited directly in the ABAP Workbench (transaction SE80). The development objects created here, such as service files, HTML templates, and MIME objects, are stored in the SAP Repository and connected to the Transport Organizer.

Outside an SAP system, developers can edit Web objects for ITS services with the **SAP@Web Studio**. Provided that there is a network connection, IAC objects can be transferred to an SAP ITS ("published") and stored in the repository of an SAP development system ("source control" with check-in and check-out). The SAP@Web Studio can be operated only on PCs with a Windows operating system.



Note: SAP@WebStudio is recommended only for use with SAP systems with Basis Release 4.6B and earlier. As of SAP Basis 4.6C, you should use the development environment within the SAP system.

An extensive tutorial is available in the online documentation for SAP NetWeaver **04** at *SAP NetWeaver → Application Platform → ABAP Technology → ABAP Workbench (BC-DWB) → ABAP Workbench: Tools → Web Application Builder for ITS-Services → Tutorial: Implementing Web Applications*.

Internet Application Components Object Receiver (IACOR)

As previously mentioned, SAP@Web Studio allows IAC objects to be published to an SAP ITS. However, to ensure that data is distributed consistently across the system landscape, you must distribute your custom IAC objects using the transport system of your SAP system (regardless of the development tool used).

You must install the IAC Object Receiver (IACOR) before you publish IAC objects to an ITS directory from an SAP system. For a virtual ITS instance, the IACOR distributes the objects transferred from the SAP system to the correct location in the file system of the ITS instance. When you install the IACOR, two RFC destinations are generated in the SAP system for each ITS instance that is supplied by the IACOR. These RFC destinations are used for publishing in the AGate and WGate. You can combine several AGates and W Gates into a "site", for example, as part of load balancing.

Watchdog

The watchdog runs as a Windows service on the host with the Web server. It offers:

- Monitoring of all local ITS instances by DCOM
- High availability of the WGate using Microsoft WLN
- Registration of the ITS on a directory service (LDAP server)

SAP ITS Monitoring

You can also use a special agent (*SAPCCMSR*) to monitor an SAP ITS using CCMS analysis monitors (transaction RZ20). For more information, see the *System Monitoring and Alert Management* area in the SAP Service Marketplace under the Quick Link */systemmanagement*, as well as SAP Note 418285.



Lesson Summary

You should now be able to:

- Describe the architecture of the SAP ITS (standalone)
- Perform simple administrative tasks on an SAP ITS

Related Information

- Online documentation for SAP NetWeaver **04** under *SAP NetWeaver* → *Application Platform* → *ABAP Technology* → *UI Technology* → *Web UI Technology* → *Internet Transaction Server* → *SAP ITS Standalone*
- Quick Link */sap-its* on the SAP Service Marketplace
- SAP Note 325616: *SAP ITS System Requirements*
- SAP Note 197746: *SAP ITS Maintenance Strategy*
- SAP Note 531617: *New Functions in SAP ITS 6.20*
- SAP Note 720428: *General Information about SAP ITS 6.20 Configuration*
- SAP Note 491781: *Corrections to SAP ITS 6.20*
- SAP Note 710041: *Collective Note: Security in SAP ITS*
- SAP Note 688295: *WGate configuration tool*
- SAP Note 651581: *Information on Load Tests*

Lesson: Internet Communication Manager

Lesson Overview

In this lesson, you will learn about the Internet Communication Manager (ICM) process and administration options.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the implementation area of the ICM
- Configure and monitor the ICM

Business Example

As part of the conversion to a modern, service-oriented IT infrastructure, new SAP applications based on Business Server Pages and SOAP services are implemented in your company. As a member of the system administration team, it is your task to configure the SAP systems in accordance with your requirements. You therefore require an overview of the central process of intranet and Internet connection – the Internet Communication Manager (ICM).

Architecture of the ICM Process

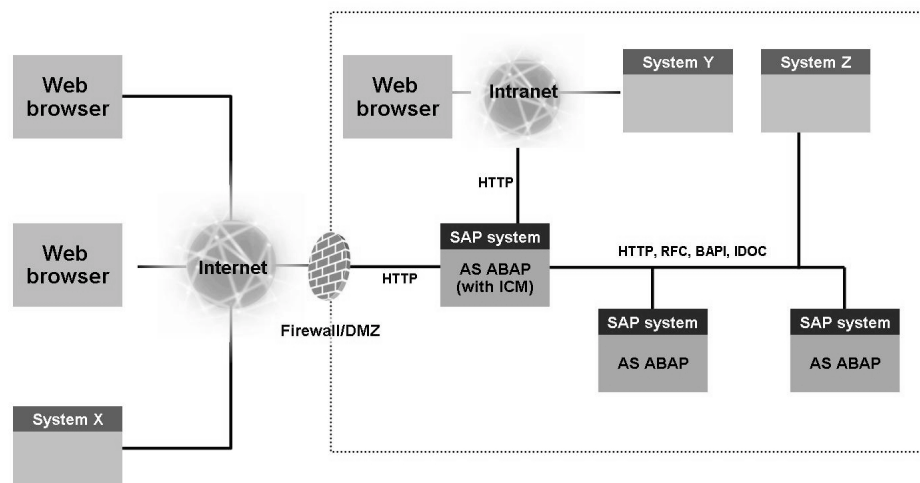


Figure 12: System Landscape with AS ABAP (example)

The figure above shows an example of a **system landscape** in which Web browsers from the Internet and intranet are connected with an SAP Web AS (in this case, distributed across a number of servers). Important features are:

- Support for standard Web protocols such as HTTP, HTTP, WebDAV, SOAP, and SMTP
- Display of standard Web formats such as HTML, XML, and XSLT
- Complete integration into the SAP environment (development environment, user administration, authorization concept, system monitoring, and communication protocols)

As of Release 6.10, the AS ABAP can function both as a **Web server** (server role) and as a **Web client** (client role). The server role, in which the SAP Web AS can accept and process HTTP requests from any Web client (such as a Web browser) and send back an HTTP response, is what we will discuss in this lesson.

Within a work process, the Internet Communication Framework (ICF) provides the environment for handling HTTP requests. The ICF is the bridge between the C kernel of the SAP system and the application program created in ABAP.

As of SAP Web AS 6.10, work processes can directly generate Web-compatible content in a way that can be forwarded to a browser using the ICM. One way of creating content of this type is to use applications with Business Server Pages (BSPs) that were developed in the SAP system using a tool of transaction SE80, the Web Application Builder for BSPs.

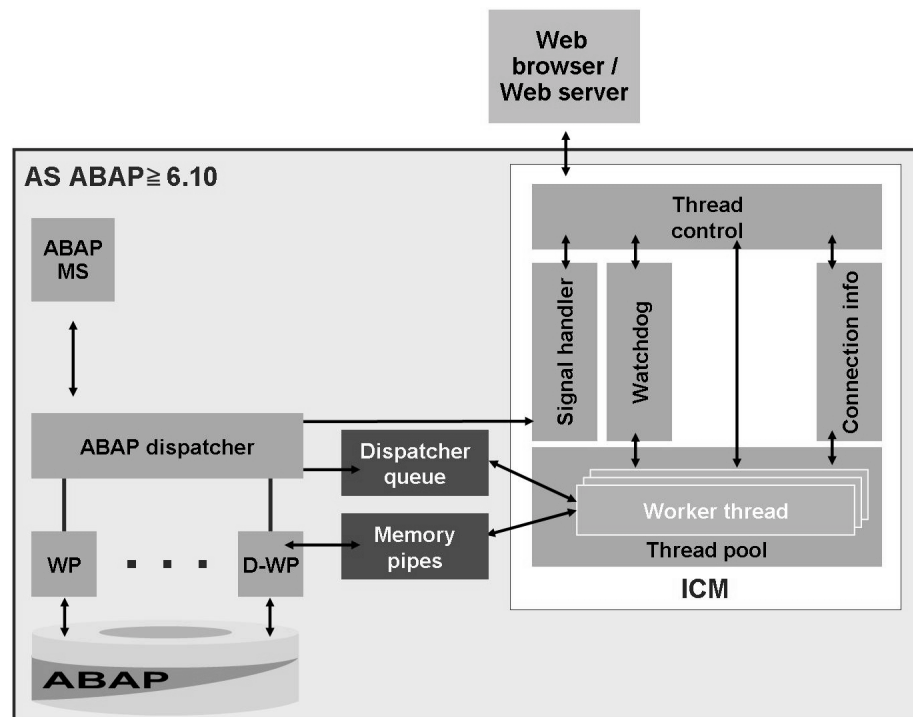


Figure 13: Internal Structure of the ICM Process

From a technical point of view, the ICM is a **separate** process (*icman* at operating system level) that is started and monitored by the ABAP dispatcher. Its task is to ensure that the SAP system can communicate with the outside world (using HTTP, HTTPS, and SMTP). In the server role, it can process requests from the Internet that arrive with URLs with the server/port combination for which the ICM is listening.

The ICM then calls the appropriate local handler, depending on the URL. The ICM process uses **threads** to process the created workload in parallel. The components of the ICM are:

- **Thread Control:** This thread accepts the incoming TCP/IP requests and creates (or raises) a worker thread from the threadpool to process the request.
- **Worker Thread:** This thread handles requests and responses for a connection. A worker thread contains an I/O handler for the network input and output, and various plug-ins for the different supported protocols.
- **Watchdog:** A worker thread usually waits for a response (whether it is client or server); if a timeout occurs, the watchdog takes over the task of waiting for the response. The worker thread can then be used for other requests.
- **Signal Handler:** Processes signals that are sent from the operating system or another process (such as the ABAP dispatcher).
- **Connection Info:** Table with information for each existing network connection.
- **Memory Pipes:** These memory-based communication objects allow data transfer between the ICM and the ABAP work processes.

The ICM uses plug-ins to implement the different communication protocols. Once the AS ABAP has been installed, the following protocols can be used immediately:

- HTTP
- HTTPS
- SMTP

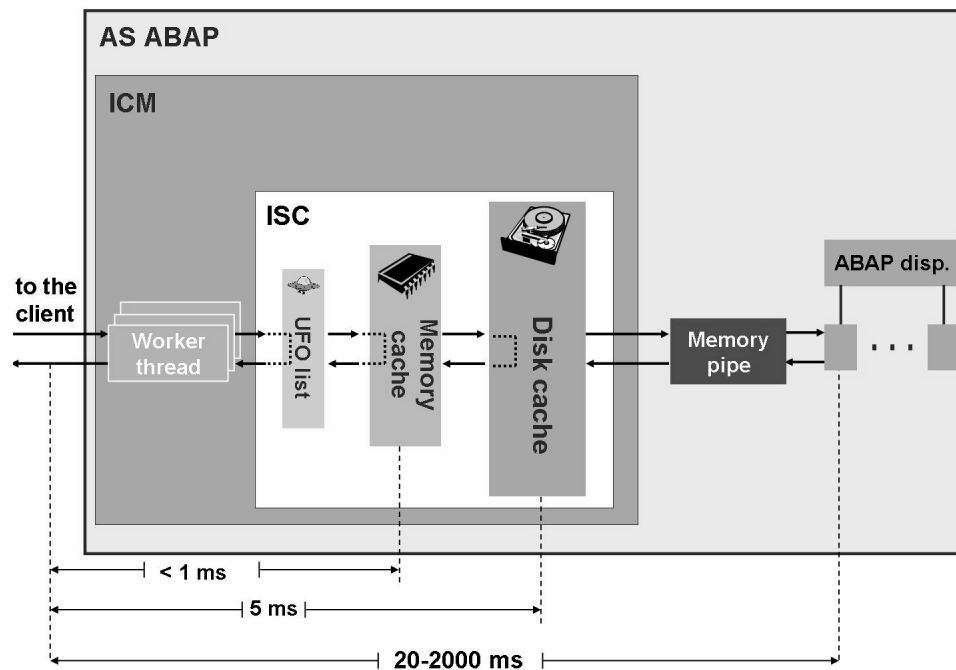


Figure 14: Internet Server Cache (ISC)

A part of the ICM that is important for performance is the Internet Server Cache (ISC), which stores HTTP(S) objects before they are sent to the Web browser. The next request can then be made directly from the ISC, provided that the expiry time has not elapsed. This avoids branching to the ABAP work process, which can accelerate access considerably.

Some features of the ISC:

- **Two-level hierarchy:** When objects are stored, the advantages of both the high speed of main memory (memory cache) and the storage capacity of hard disks (disk cache) are used.
- **Dynamic Caching:** Traditional products are based on HTTP proxies and usually offer caching only of static content, such as images. The ISC can also cache dynamic content such as JSPs or BSPs.
- **Active Caching:** The application has full control over ensuring that the objects in the cache are up to date.
- **UFO Caching:** Invalid requests (“UnFound Objects”) that lead to error situations in the application server or the database are directly rejected, so that the system is protected against invalid or malignant requests.
- **Browser-dependent Caching:** The developer of a BSP can define whether his or her application is dependent on browser type. If this indicator is set, the ISC uses the data in the cache only for requests from the same browser type.

The ISC is configured using the profile parameter `icm/HTTP/server_cache*` and can be monitored and invalidated from the SAP system.

Start Procedure and Monitoring



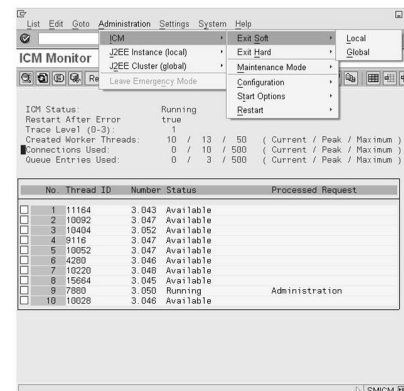
System Start

Evaluation of the profile parameter

`rdisp/start_icman`

Possible values: *true* or *false*
Default: *true*

Runtime



ICM Monitor, transaction *SMICM*

Figure 15: Starting the ICM

The profile parameter `rdisp/start_icman` controls whether an ICM process is also started when an application server is started. If no value is specified, the default setting *true* applies. You configure the ICM using profile parameters (most of which begin with `icm/`). The settings for `icm/server_port_<xx>` are of particular importance. These settings determine the port used for each protocol, as well as other attributes of the protocol (such as timeout).

In the SAP system, you can quickly obtain an overview of which application servers are running with an ICM using the server overview (transaction SM51).

For more detailed information (such as the thread ID), see the **ICM monitor** (transaction SMICM). From this transaction, you can choose the menu path *Administration* → *ICM* to terminate the ICM with a soft termination (corresponds to Unix signal 2) or a hard termination (corresponds to Unix signal 9). The dispatcher then starts a new ICM process. By choosing *Administration* → *ICM* → *Restart* → *Yes/No*, you can control whether the ABAP dispatcher restarts the ICM if it was terminated by an error or at the request of an administrator.

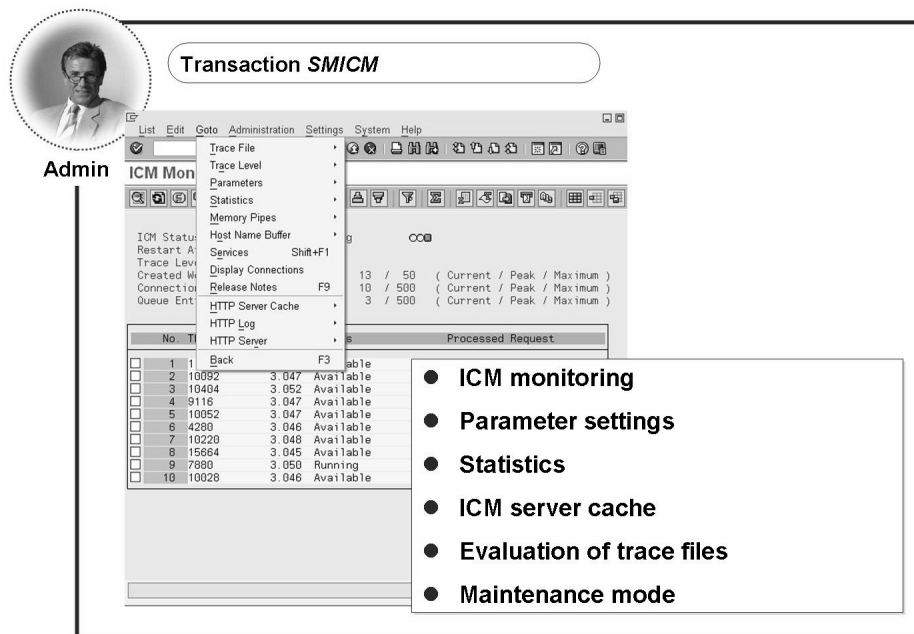


Figure 16: ICM Monitor Functions

The most important tool for an administrator in the ICM environment is the ICM monitor (transaction SMICM). Note that the data displayed is **instance-dependent** (in the same way as the work process overview SM50). Some administrative activities (all available from transaction SMICM) are:

- Monitoring and restarting the ICM
- Configuring the trace level (*Goto → Trace Level → ...*), values from 0 to 3.
- Evaluating the trace files (*Goto → Trace File → ...*); the system reads the *dev_icm* file from the *work* directory of the current instance.
- Overview of the profile parameters (*Goto → Parameters → Display/Change*). The ICM is configured using profile parameters. The displayed values apply for the instance to which you are currently logged on. For documentation on the parameters, see the ICM monitor (*Goto → Parameters → Change* and choose *Documentation*), transaction RZ11, and SAP Library.
- Display the statistics (*Goto → Statistics → Display*). You can use these statistics to find out how many requests the ICM has processed since it was started (or since the statistics were reset). The system also displays information about processing duration.
- Monitoring (*Goto → HTTP Server Cache → Display*) and resetting (*Goto → HTTP Server Cache → Invalidate → ...*) the ICM server cache. The ICM server cache stores HTTP objects before they are sent to the client. The next time that this object is requested, the content can be sent directly from the cache to the client.
- In maintenance mode, the ICM logs off from the ABAP message server and is not available for Web requests. The ICM processes only the remaining requests. If an Internet user accesses an ICM in this status from the browser, the system issues a message stating that the ICM is in “Maintenance Mode”.

You can determine some of the listed data at operating system level using the *icmon* program. The call *icmon -h* displays the possible parameters for this small program, which can also, among other things, generate requests to simulate normal system workload.

Exercise 1: Administration of the ICM

Exercise Objectives

After completing this exercise, you will be able to:

- Monitor the ICM process

Business Example

As part of SAP BI, your company uses browser-based functions such as Web reporting, interactive charts, and the Business Explorer Browser (BEx Browser). As an administrator, you are responsible for monitoring the ICM processes that establish the connection between the Web browser and the SAP system.

Task 1: Checking the ICM Settings

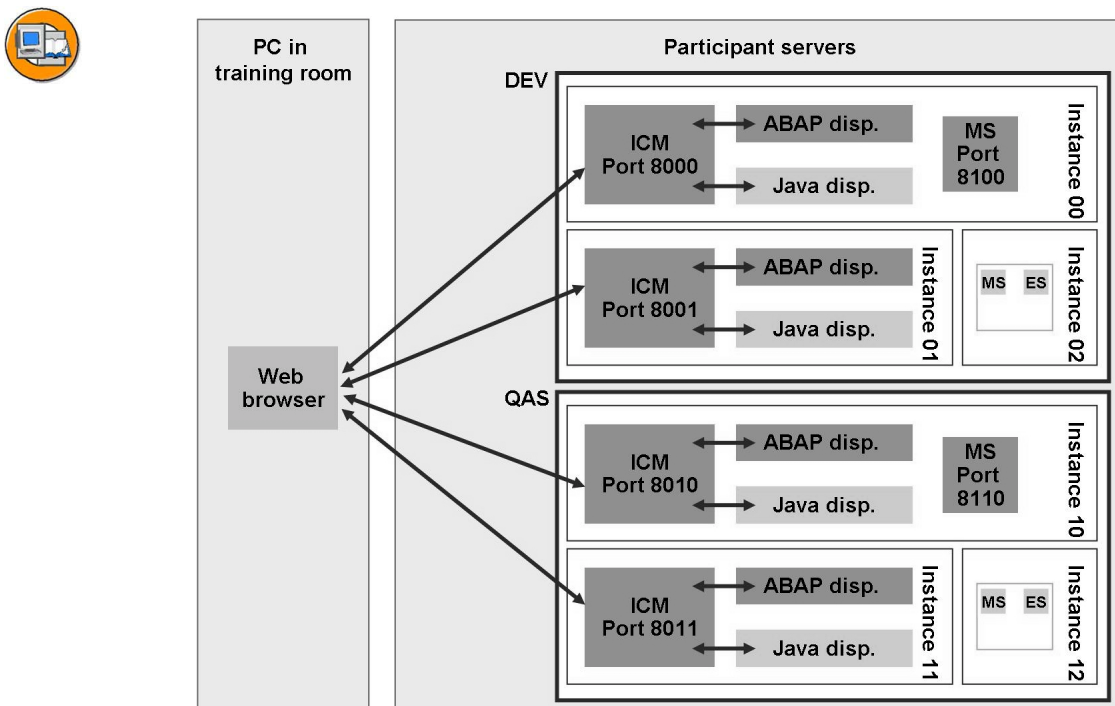


Figure 17: Complete Scenario of the Training Landscape

Number, port, and release of the ICM processes in the training environment

1. How many ICM processes are running in your SAP system?

Continued on next page

2. Determine the port through which requests in the HTTP protocol are processed for the application server to which you are currently logged on.
3. Which release of the ICM is used on the training system?

Result

You know the port and release for the ICM process on the training system.

Task 2: Simple HTTP Requests

Start a request in the Web browser and monitor it with the ICM monitor.

1. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/public/ping`** (example for group QAS and server twdf0042: **`http://twdf0042.wdf.sap.corp:8010/sap/public/ping`**).

The message “Server reached successfully” appears.



Note: The ICM port may not be available from your training room. In this case, use the Web browser available on your server.

2. Open the ICM monitor and note how many requests from your Web browser (you may have to *Refresh* the URL above) have been processed by the worker threads.



Hint: The data in the ICM monitor is instance-specific.

3. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/public/icman/ping`**.
4. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM Port>/sap/public/icman/mime/theme.jpg`**.

Result

You can monitor the activity of the ICM.

Continued on next page

Task 3: Load Test with icmon Tool

Monitor the ICM work threads under a generated workload.

1. At operating system level of your server, start the **icmon pf=<instance profile name>** command and enter an instance profile of your SAP system for **instance profile name**.

Call the menu (by pressing **m**) and generate load with the following values:

<i>Host</i>	Host on which your SAP system is running, such as twdf0042.wdf.sap.corp (default setting)
<i>Port</i>	An ICM Port valid for your system, such as 8011
<i>'l.x'=HTTP/l.x or '9.x' HTTPS</i>	1.0 (default setting)
<i>Get request data from file</i>	No (default setting)
<i>Path</i>	/sap/public/icman/mime/theme.jpg
<i>Optional Attributes</i>	No (default setting)
<i>Expected OK Code</i>	0 (default setting)
<i>Think time in millisecs</i>	0 (default setting)
<i>Number of requests</i>	5000
<i>Number of threads</i>	10

2. Observe in the ICM monitor how the requests generated by *icmon* are processed by the worker threads.

Result

You can use the *icmon* tool to monitor the ICM and to start workload simulations.

Solution 1: Administration of the ICM

Task 1: Checking the ICM Settings

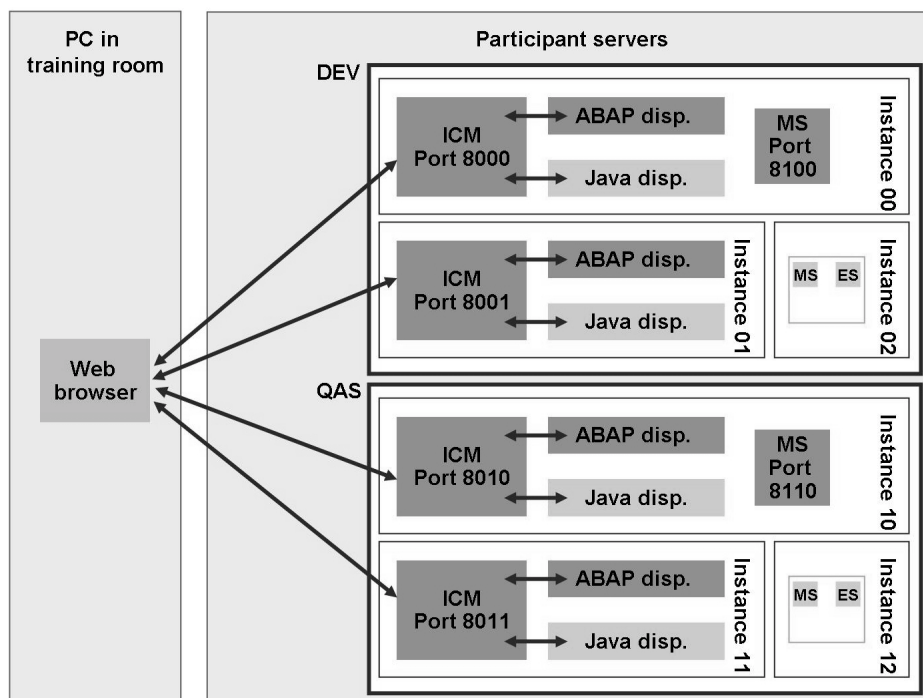


Figure 18: Complete Scenario of the Training Landscape

Number, port, and release of the ICM processes in the training environment

1. How many ICM processes are running in your SAP system?
 - a) In the server overview (transaction SM51), count the application servers for which the ICM process is listed. An ICM process should be configured for each of your ABAP instances.

Continued on next page

2. Determine the port through which requests in the HTTP protocol are processed for the application server to which you are currently logged on.

- a) Check the value of the profile parameter *icm/server_port_0*, for example:
 - In the ICM Monitor (transaction SMICM) by choosing *Services* or *Goto* → *Services*
 - In the ICM Monitor (transaction SMICM) by choosing *Goto* → *Parameters* → *Display*
 - By executing report *RSPFPAR*
 - In transaction RZ10



Hint: The determined port is instance-specific. In the training systems, the parameter *icm/server_port_0* has the value *PROT=HTTP,PORT=80\$\$*. The variable *\$\$* is replaced by the instance number when the ICM is started, ensuring that ports are unique in all cases.

3. Which release of the ICM is used on the training system?

- a) You can determine the ICM release in the ICM monitor by choosing *Release Notes* or *Goto* → *Release Notes*.

The information that you are looking for is at the start of the list. At the end of the list, all problems that are solved with the current patch level are listed (with associated SAP Notes).

Result

You know the port and release for the ICM process on the training system.

Task 2: Simple HTTP Requests

Start a request in the Web browser and monitor it with the ICM monitor.

1. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/public/ping`** (example for group QAS and server twdf0042: **`http://twdf0042.wdf.sap.corp:8010/sap/public/ping`**).

Continued on next page

The message “Server reached successfully” appears.



Note: The ICM port may not be available from your training room. In this case, use the Web browser available on your server.

- a) Enter the specified URL in your local Web browser and choose *Enter*.



Hint: All services under */sap/public* use a predefined user; therefore, no logon is required for this request.

As of AS ABAP 6.20, services must be explicitly activated. This has already been done in the training system for the services specified in this exercise.

2. Open the ICM monitor and note how many requests from your Web browser (you may have to *Refresh* the URL above) have been processed by the worker threads.



Hint: The data in the ICM monitor is instance-specific.

- a) In the ICM monitor (transaction SMICM), choose the *Refresh* button after you have sent a few requests to the ICM.
3. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/public/icman/ping`**.
 - a) See task description. The message “server on host twdfXXXX system twdfXXXX_<SID>_<Instance> (000) successfully reached ” appears.
4. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM Port>/sap/public/icman/mime/theme.jpg`**.
 - a) See task description. A small icon should appear.

Result

You can monitor the activity of the ICM.

Continued on next page

Task 3: Load Test with icmon Tool

Monitor the ICM work threads under a generated workload.

1. At operating system level of your server, start the **icmon pf=<instance profile name>** command and enter an instance profile of your SAP system for **instance profile name**.

Call the menu (by pressing **m**) and generate load with the following values:

<i>Host</i>	Host on which your SAP system is running, such as twdf0042.wdf.sap.corp (default setting)
<i>Port</i>	An ICM Port valid for your system, such as 8011
<i>'l.x'=HTTP/l.x or '9.x' HTTPS</i>	1.0 (default setting)
<i>Get request data from file</i>	No (default setting)
<i>Path</i>	/sap/public/icman/mime/theme.jpg
<i>Optional Attributes</i>	No (default setting)
<i>Expected OK Code</i>	0 (default setting)
<i>Think time in millisecs</i>	0 (default setting)
<i>Number of requests</i>	5000
<i>Number of threads</i>	10

- a) If you have not already done so, log onto the Terminal Services Client (also known as the RDP Client) at operating-system level using the user **<sid>adm**.
- b) Open Windows Explorer and navigate to the directory **G:\usr\sap\<SID>\SYS\profile**. Click the *profile* directory with the secondary mouse button and choose *Command Prompt Here*.
- c) Start the *icmon* program by specifying an instance profile: **icmon pf=<instance profile name>** (example for the dialog instances of group QAS on server twdf0042: **icmon pf=QAS_D11_twdf0042**).
- d) Enter the command **m** to switch to the *Monitor Menu* and generate load by entering the command **G** and the parameters specified in the task description.

Continued on next page

2. Observe in the ICM monitor how the requests generated by *icmon* are processed by the worker threads.
 - a) In transaction SMICM, choose the *Refresh* function to observe the activity of the worker threads. You may notice that the ICM starts further worker threads.

Remember that the display in transaction SMICM is not system-wide, but only applies to your instance.



Note: The default load data generates 50,000 requests (that is, requests multiplied by threads).

Result

You can use the *icmon* tool to monitor the ICM and to start workload simulations.



Lesson Summary

You should now be able to:

- Describe the implementation area of the ICM
- Configure and monitor the ICM

Related Information

- Online documentation of SAP NetWeaver 7.0 under *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *Platform-Wide Services* → *Architecture of the SAP NetWeaver Application Server* → *Internet Communication Manager (ICM)* and *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Solution Life Cycle Management by Key Capability* → *System Management* → *Administration of the Internet Communication Manager*.
- *SAP NetWeaver* → *Application Platform* → *Architecture of the SAP Web AS* → *Internet Communication Manager (ICM)* and *SAP NetWeaver* → *Solution Life Cycle Management* → *System Management* → *Administration of the Internet Communication Manager*.
- SAP Note 851852: *ICM Patch Collection (7.00)*
- SAP Note 737625: *Parameter Recommendations for the ICM*
- SAP Note 421359: *ICM Connection of Ports <1024 to Unix*
- SAP Note 634006: *Information for Explaining ICM Messages*

Lesson: Internet Communication Framework

Lesson Overview

The Internet Communication Framework (ICF) provides an environment for handling Web requests in the ABAP work process of an SAP system. This lesson introduces the ICF and provides more information about some administrative issues. The last section is dedicated to the integrated ITS as of SAP Web AS 6.40.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Activate and use the integrated ITS as of AS ABAP 6.40

Business Example

Your company wants to use Web applications based on Web dynpro, BSPs, or the integrated ITS to connect your SAP systems with the Internet. As a member of the system administration team, it is your task to create links between called URLs and services and programs of the SAP system.

Classifying the ICF

The Internet Communication Framework (ICF) provides a way for different systems to communicate with each other over the Internet using standard protocols (such as HTTP and SMTP). No additional programming libraries (for AS ABAP) are required from SAP. However, for the HTTPS protocol, the SAP Cryptographic Library (SAPCRYPTOLIB) must be installed and configured (see SAP Note 510007). Your system platform only must be configured to be Internet capable. This scenario allows for the most flexible setup of the overall communication requirements.

The ICF allows a response to a request to be generated using an application. An HTTP request is sent from a client (such as a Web browser) to the server. It is then forwarded to an application by the ICF. Here, data is collected and sent back to the client as a response by the ICF. The response data is then displayed in the browser.

The following provides more information about using the SAP system as a Web server (HTTP(S) server). For information about the Web client role of the SAP system, see the online documentation.

The application logic that is to be called by an HTTP request from the intranet or Internet is implemented by the **HTTP request handler** in each case. An HTTP request handler is a program (or, more precisely, an ABAP class) that is identified using a URL, and which receives HTTP requests that use this URL. The task of the HTTP request handler is to receive the data that is sent by a request (for example, coded into the URL as “query string” information), to perform a number of handler-specific processes, and to generate a response to this HTTP request.

Customers can also create these HTTP request handlers themselves, although SAP does provide some. The most commonly used SAP HTTP request handler is that for the Business Server Pages (BSP) and can be used to develop simple Web applications.

If an HTTP request is received by the ICM that is to be processed in a work process, the task handler takes control. It then starts the ICF controller. From now on, we are in the ABAP world and in the ICF.

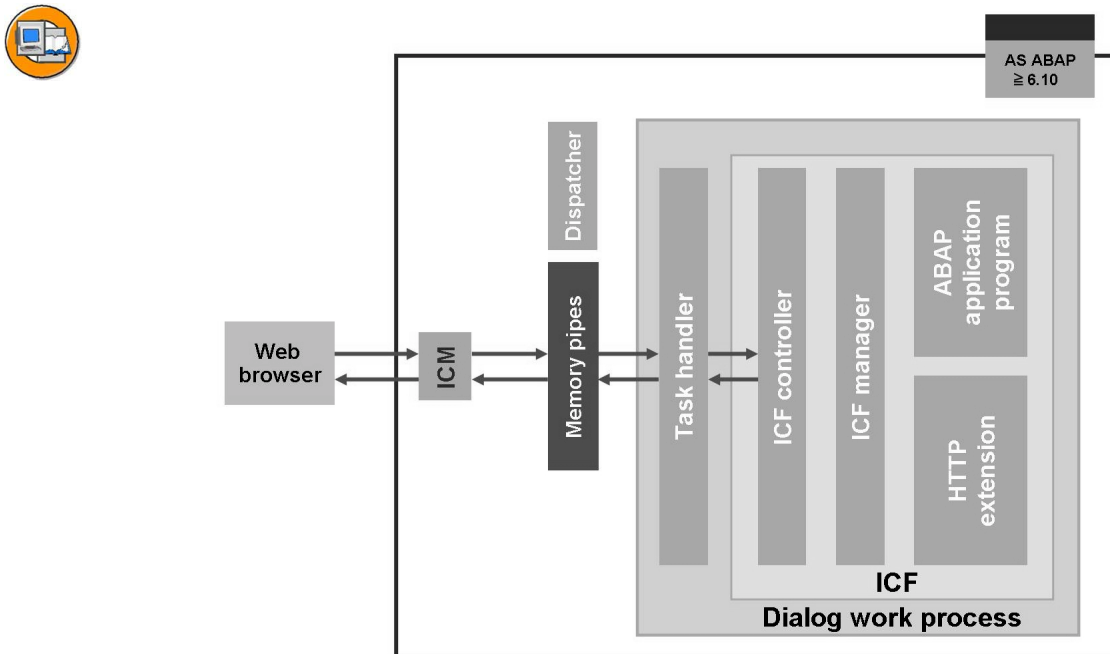


Figure 19: Interaction Model of an SAP System in the Server Role

An HTTP(S) request is processed in the following steps:

1. The request is sent from the user's Web browser to the ICM using the HTTP protocol. The ICM uses the requested URL to determine whether the application called is implemented in the ABAP or Java stack of the SAP NetWeaver Application server.

This example uses an ABAP application that must be processed by a dialog work process.

2. The ICM stores the data received in a memory pipe (in the shared memory) and informs the ABAP dispatcher.
3. The ABAP dispatcher adds the ICM request to the dispatcher queue, creates a new context (if there is no context that is processed statefully), and selects a work process for processing.
4. The task handler in the work process reads the data from the memory pipe and transfers it to the ICF controller, which is implemented using function module *HTTP_DISPATCH_REQUEST*.
5. The ICF controller transfers the request to the ICF manager, which is implemented by the ABAP class *CL_HTTP_SERVER*. The ICF controller creates a server control block and fills it with the HTTP request data that it requested from the ICM.
6. The client is then authenticated, whereby several logon options are available.
7. The HTTP request handler determined previously is called (this can process the request data, call further applications, access the response object, and so on). When the HTTP request handler is ready, it returns control to the ICF controller.
8. The task handler writes the response back to the memory pipe (response serialization) and signals to the ICM that it has finished processing the request.
9. The ICM returns the response to the Web browser.

Properties and Maintenance of ICF Services

From a technical point of view, there is an ABAP class behind an HTTP request handler. This class implements the interface *IF_HTTP_EXTENSION* and the method *HANDLE_REQUEST ()*. SAP delivers classes of this type; customers can, of course, also create their own classes with the Class Builder (transaction SE24, integrated into the Object Builder, and transaction SE80).

Linking a particular URL with an HTTP request handler is the task of ICF services. An ICF service therefore creates a connection between a URL to which an HTTP request is sent and development objects that process this request.

An SAP system (with AS ABAP its technical foundation) already contains various services when it is delivered. The exact scope depends of course on the system type (SAP ECC, SAP CRM, and so on) and the release. You can obtain an overview of all available services using the central maintenance transaction for ICF services, transaction SICF. All available services are displayed in a hierarchical structure in transaction SICF. The complete path for a service (such as `/sap/bc/icf/info`) ultimately determines (together with the protocol, server name, and port) the URL under which the service can be called. The following section explains some of the aspects that are relevant for administrators in more detail.

Activation concept

ICF services can be active or inactive, which is indicated by different colors in transaction SICF:



Status of ICF Services

Status	Color in SICF	Meaning
Active	Black	Service can be called
Inactive	Gray	Service explicitly deactivated
	Blue	Service implicitly deactivated

For implicitly deactivated services, there is always a higher-level service in the ICF tree that has been explicitly deactivated. If you activate this service (displayed in gray), all lower-level services that were implicitly deactivated (displayed in blue) are activated. When you activate a node (by choosing *Service/Host* → *Activate* or using the context menu that appears when you click the secondary mouse button), you can choose whether you want to explicitly active the selected service only (*Yes*) or all of the lower-level subservices (*Yes* with the tree icon).

If you try to call an inactive service, the system displays a message stating that access to this page is blocked. Activated ICF services are a security risk since they can be accessed directly using HTTP(S) or SMTP from the intranet or Internet (depending

on your network configuration). You should therefore restrict access using suitable measures, such as by activating only the required ICF services and assigning the relevant authorizations to users.



Hint: At the time of delivery, all ICF services are inactive so that no ICF service can initially be used.

If SAP provides changes to ICF services as part of Support Packages, these services are also inactive when you import the Support Package (regardless of the activation status before import).

Properties and Inheritance

An ICF service is characterized by **properties** that you maintain in transaction SICF. By double-clicking a service, you access the *Create/Change a Service* screen on which you can configure the following settings:

Service Data

- An **inheritance principle** applies to the properties of an ICF service: In transaction SICF, you do not have to maintain properties for each individual service. You can do this simply for the higher service nodes (for example, */sap/bc/bsp*). All lower-level services are then assigned these properties, provided that other values have not been entered explicitly for them.

This inheritance process is not always required. You can use the *Do Not Include Inherited Settings* indicator to control whether this inheritance logic is interrupted. As of AS ABAP 7.00, you can choose *Display Inheritance* to show the properties for the current service that are inherited from higher-level ICF services.

- Under *Load Balancing*, you can enter a logon group (from transaction SMLG) using the input help (F4). When you use the SAP Web Dispatcher, requests sent to this service are forwarded only to the ABAP instances of the logon group defined.
- If you enter a value in the *SAP Authoriz.* field, the system checks whether the user has this authorization (for authorization object *S_ICF*, field *ICF_VALUE*) at runtime.
- Once the time defined in *Session Timeout* has expired, a stateful application is terminated (if the value is 0, the profile parameter *rdisp/plugin_auto_logout* has a default value of 30 minutes).
- If you set *Compression* to yes, the SAP system compresses the response (using the *gzip* process), provided that the caller can control decompression.
- If you set *GUI Link* to yes, the screen images that are generated in the application by processing conventional dynpros are converted to a format that allows them to be displayed graphically in a browser.

This function (as well as the screen that can be accessed with *GUI Configuration*) is required for the integrated ITS as of SAP Web AS 6.40, and is detailed below.

- The *Support Accessibility* indicator specifies that an accessibility mode is called if the application has one. However, it cannot be guaranteed that this is the case.

Logon Data

There are various ways for an HTTP request to log onto the AS ABAP, and you can configure these for each individual service node. With the *Standard* default setting, the following check procedures are used in exactly this sequence:

1. Fields authentication (logon using HTTP fields)
2. SSO authentication (logon using Single Sign-On)
3. Basic authentication
4. SAP authentication (logon using SAP user and password)
5. Certificate authentication (logon using client certificate)
6. Service authentication (logon using the anonymous data entered in the service)

By choosing *Alternative Logon Procedure*, you can select any logon procedure (in the *Logon Procedure List* that appears) and change the check sequence.

With *Required with Logon Data*, only those entries specified in the service under *Logon Data* (*client*, *user*, *password*, and *language*) are used for the check. You should enter only those users that were created in transaction SU01 as *service users*. If you enter a dialog user, the system issues a warning message.

If you select *Required with Client Certificate (SSL)*, logon occurs exclusively with an X.509 client certificate.

For the *Standard* and *Alternative Logon Procedure*, you can select *Use All Logon Procedures* to specify whether the respective check sequence is to run until one of the logon procedures is successful, or whether the caller is to receive a negative confirmation as soon as the first logon procedure fails.

Depending on the procedure selected, you can configure additional settings (for example, you can require SSL, that is, the https protocol).

Handler List

Here, you list the HTTP handlers in the sequence in which they are to be executed. An HTTP request handler is an ABAP class that implements the interface *IF_HTTP_EXTENSION*. This interface contains the method *HANDLE-REQUEST*, which is called by the ICF.

Error Pages

On the *Error Pages* tab page, you can specify which response pages are to be sent to the caller in the following situations:

- *Logon Errors* (HTTP 401: logon failed)
- *Appl. Errors* (HTTP 500: An error occurred in the application, for example, ABAP short dump)
- *Logoff Page*
- *Not Accessible* (HTTP 404)

In each case, either an explicit response page can be sent to the browser or the caller can be redirected to another URL. In the *Logon Errors* area, you can also enable a direct system logon if an error occurs.



Note: Under the */sap/public* node, services are defined that are required for system-internal services. These differ from the other services in the tree since no user is maintained. You therefore do not have to log onto the SAP system. The actions are carried out under the *SAPSYS* system user. Therefore, customers are not permitted to create their own services under the */sap/public* node.

Administration

The *Administration* tab page contains administration data such as the user who created and last changed a service.



Hint: If logon data is defined for an ICF service that is then transported, the logon data is deleted during transport. This is due to security reasons. Furthermore, it is not possible to guarantee that the respective user exists in the target system. The user must therefore be maintained in the target systems. For more information, see SAP Note 732218 *ICF: Logon data from SICF is not transported*. Also for security reasons, ICF services that have been transported to a target system are initially inactive and must be activate explicitly (see SAP Note 517484: *Inactive services in the Internet Communication Framework*).

Aliases

In the ICF, you can link from one ICF service to another "alias". A distinction is made between internal and external aliases:

On the *Create a Service Element* screen in transaction SICF, if you create a service and choose *Reference to Existing Service*, you create an **internal alias**. Instead of defining an HTTP request handler, use the *Alias Trgt* tab page to specify (by double-clicking) the page to which the alias is to refer in the HTTP service tree. This allows you, for

example, to call the existing and unchanged service with alternative settings (such as logon data and procedure). If possible, customers should not create internal aliases to SAP services (which are always in the */sap/* namespace).

To allow services to be called with any meaningful, non-technical names, customers should use **external aliases**. Therefore, switch to the *External Aliases* view in transaction SICF. Unlike an internal alias, an external alias can contain a forward slash (/) in its name; otherwise, both procedures are handled in the same way.

Monitoring

The **ICF recorder** enables developers and administrators to identify and correct sources of error in failed service calls by recording HTTP requests.

You can use it to save recorded requests (without the passwords used) in the system database. This facilitates the evaluation process since it is usually no longer necessary to describe the error so that the problem can be reproduced. The problem can be executed multiple times using the database entry in order to further identify the cause of the problem by debugging or work process traces. Once the problem has been corrected, the erroneous data can be used to check the corrections.

You can call the ICF recorder from transaction SICF by choosing *Edit* → *Recorder* → *Activate/Deactivate/Display Recording*; you can also use transaction SICFREORDER for the evaluation process. The basic steps are:

1. Activate the recording. You have to enter:
 - The URL path to be recorded (if you have previously selected a path, this is used as the default value)
 - The duration of the recording (*Record Time*) and storage in the database (*Lifetime*)
 - Whether the requests of one user (recommended) or all users of the current client are to be recorded
2. Call the services to be monitored (if necessary, using the selected user).
3. Deactivate the recording (to prevent performance losses).
4. Display and process the recorded requests.

In the administrator settings (available in transaction SICF under *Goto* → *Settings*), you can prevent the ICF recorder from being used system-wide. You use authorization object *S_SICFREC* to control access to the request data using the ICF recorder.

AS ABAP with Integrated ITS

SAP ITS as a standalone software component (comprising the Web server, WGate, and AGate) is also considered a “standalone” version and is supplied by SAP up to and including ITS Release 6.20. With Release 6.40 of the AS ABAP, the SAP ITS was integrated in the ABAP kernel (under the name “SAP Integrated ITS”). This means that as of SAP Web AS 6.40, it is not necessary to install separate ITS components or servers.



Hint:

- For AS ABAP 6.20 and earlier, only the SAP ITS standalone 6.20 can be used (and is supported).
- For AS ABAP 6.40, both the integrated ITS and an SAP ITS 6.20 standalone is released.
- As of AS ABAP 7.00, only the integrated ITS can be used (and is supported).

Architecture

The integrated ITS is completely integrated in the familiar infrastructure of the AS ABAP: It is accessed via the ICM process, implemented as an ICF service, and uses the database as an object storage location.

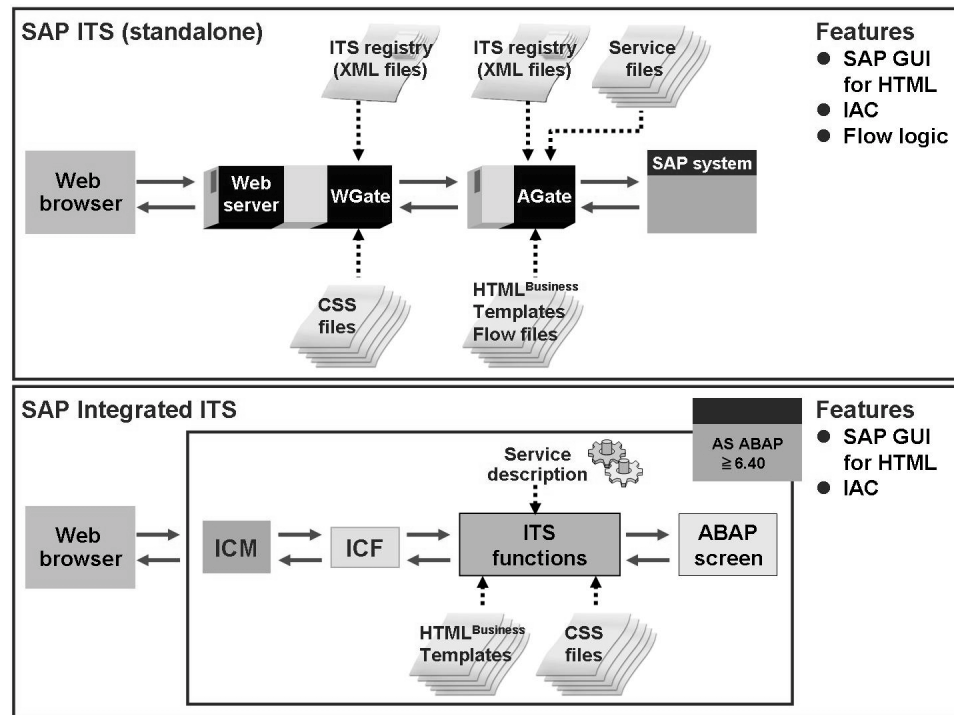


Figure 20: Architecture Comparison: SAP ITS (Standalone) and SAP Integrated ITS

This setup has the following advantages:



- A separate Web server and ITS server is no longer required. By reducing the administration and maintenance expenses, the **total cost of ownership** (TCO) is also reduced.
- The ITS is available on all platforms released for the AS ABAP, which significantly enhances the **platform matrix** (see service.sap.com/pam on SAP Service Marketplace).
- With the standalone ITS, some customers deliberately separate the WGate and AGate to create a firewall. This **security attribute** is also available in the integrated ITS by placing the firewall between the SAP Web Dispatcher and the ICM process.
- Administrators do not require special **administration tools** for configuration and monitoring purposes.
- Developers do not have to **publish** applications on external servers; IACs are accessed via a “pseudo-publish operation” from the system database to the *INTERNAL* site.



Note: You will learn about the SAP Web Dispatcher later in this unit.

It is technically possible to operate a standalone SAP ITS 6.20 in conjunction with an SAP Web AS 6.40 (but not later releases). This may result from one of the following restrictions in comparison to the standalone version:



- The integrated ITS does not support the programming models Flow Logic, WebRFC, or WebReporting. Applications that use these still require SAP ITS 6.20.
- The integrated ITS always runs on the SAP system of which it is part. This means that a standalone ITS is required for SAP systems based on SAP Web AS 6.20 or lower.
- Only by upgrading to AS ABAP 6.40 can you ensure that a productive standalone ITS is not transferred to an integrated ITS. Depending on the initial scenario, further steps may be required (transfer of programming objects and settings).



Note: For migration guidelines for Internet applications that were originally written for the SAP ITS standalone, see the online documentation for SAP NetWeaver 7.0 under *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *ABAP Technology* → *UI Technology* → *Web UI Technology* → *ITS/SAP@Web Studio* → *SAP ITS in the SAP Web Application Server* → *Developing IACs with the Integrated SAP ITS* → *Migration....*

Configuration

The integrated ITS is automatically installed with the SAP kernel as part of AS ABAP 6.40. To use the integrated ITS, the following prerequisites must be met:



- The ICM process is operational and configured for HTTP(S).
- Profile parameter *itsp/enable* is set to **1**.
- The required ITS service is published to the *INTERNAL* site.
- The required ITS service is active in the ICF and the property *GUI Link* is set to *Yes*.
- The ICF service */sap/public/bc/its/mimes* is active in the ICF and the property *GUI Link* is set to *Not Specified*.

For the integrated ITS, various **profile parameters** are relevant, which all begin with *itsp/*. Administrations can use the usual methods to call documentation for individual parameters (transaction RZ11) and change the assigned values permanently (transaction RZ10).

Two profile parameters are particularly significant in relation to the integrated ITS:

- *itsp/enable*: You use this to deactivate (**0**) and activate (**1**) the integrated ITS. Even if the integrated ITS is active, it uses the system resources only when it is actually used. However, it can be useful to deactivate it for selected instances so that no users can access the SAP system via the SAP GUI for HTML with these instances (for example, batch or update instances). Since the conversion of SAP screen images to HTML pages also requires CPU time, it is useful to reserve a number of dedicated application servers for use with SAP GUI for HTML and to use a special logon group for load balancing between them.
- *em/global_area_MB*: This parameter determines the memory commonly used by all ABAP work processes of the SAP kernel. The integrated ITS uses it for session information and the runtime version of the HTML business templates. The required memory space depends on the number of sessions currently in use, as well as the number and size of the templates used when users want to call and display services. If your users log onto the ITS in different languages or with different browsers (for example, Microsoft Internet Explorer and Netscape Navigator), or if you require additional services not included in SAP GUI for HTML, the number of templates used increases and you will have to modify *em/global_area_MB* (see SAP Note 742048).

Alongside these profile parameters, which are evaluated by the kernel and affect the entire integrated ITS, there are also **service parameters**, which affect the individual ITS services. You maintain these settings in transaction SICF.

The basic behavior (such as logon, anonymous logon data, service options, security requirements, basic authorizations, and customized error pages) results from the properties of the ITS services in the ICF, as is also the case with “normal” ICF services. You maintain other, ITS-specific service parameters (that begin with ~) in transaction SICF on the *Create/Change a Service* screen. Choose the *Service Data* tab page and in the *Interactive Options* area, choose *GUI Configuration*. For more information about these parameters, see the online documentation for SAP NetWeaver 7.0 under *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *ABAP Technology* → *UI Technology* → *Web UI Technology* → *ITS/SAP@Web Studio* → *SAP ITS in the SAP Web Application Server* → *Configuration* → *Parameters for Administrators/Developers*.



Note: The *global.srv*, *webgui.srv*, and *<Servicename>.srv* files and their service parameters used with the standalone ITS do not exist in the integrated ITS.

Developers create new ITS services with the *Web Application Builder for ITS Services*, a tool in the ABAP Development Workbench SE80. Services are published to the (implicitly available) *INTERNAL* site. The ITS service must also be created in the ICF, and can be accessed with the URL **http(s)://<server with domain>:<ICM port>/<ICF path>/<ITS service name>** (or by choosing *Test Service* in transaction SICF).

SAP GUI for HTML with the Integrated ITS

In numerous installations, customers activate the integrated ITS as of AS ABAP 6.40 so that they can comfortably use the SAP GUI for HTML. In addition, several IACs use objects of the SAP GUI for HTML, which therefore have to be activated.

In addition to the general settings listed above, the following prerequisites also apply to the SAP GUI for HTML:



- The Internet services *system* and *webgui* are published to the *INTERNAL* site.
- The ICF service */sap/bc/gui/sap/its/webgui* is active in the ICF and the property *GUI Link* is set to **Y**.



Hint: For detailed information about the storage structure in the integrated ITS, see SAP Note 678904: *ITS - New storage structures as of SAP Web AS 6.40*.


You call the SAP GUI for HTML by entering the URL `http(s)://<server with domain>:<ICM port>/sap/bc/gui/sap/its/webgui`. The above comments about the properties of an ICF service (such as client, logon language, and so on) also apply here.

Monitoring

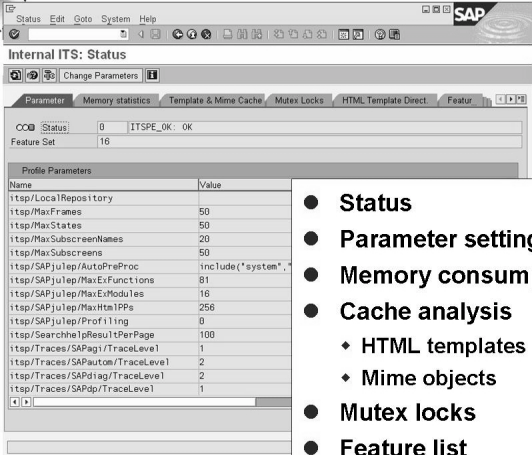
The administrator can use the tools integrated in the AS ABAP (such as transactions SM21, ST22, SMICM, and SICF) to monitor the integrated ITS. The integrated ITS does not have special trace files, but instead uses the standard developer trace files of the work processes `dev_w*.trc`. Developers can specially activate ITS tracing (in transaction SM50 or with report `RSTRC000`, component `W` for *WebGui*).

They can also use transaction SITSPMON and program `SITSPMON`, which provide a detailed status summary of the integrated ITS.





Transaction or Program *SITSPMON*



Admin

- **Status**
- **Parameter settings**
- **Memory consumption**
- **Cache analysis**
 - ◆ HTML templates
 - ◆ Mime objects
- **Mutex locks**
- **Feature list**
- **Status**

Figure 21: Monitor Functions for the Integrated ITS

The following list contains a number of selected monitor functions for the integrated ITS:

- Status: message text, FeatureSet version, and parameters
- Memory consumption: Overview and details about memory consumed by sessions and ABAP work processes
- Caches: Status and invalidation of caches for HTML templates and MIME objects
- Mutex locks (from “mutual exclusion”: technology for preventing simultaneous access to a resource by several processes)

New features of the integrated ITS have been added with AS ABAP 7.00. These include an improved use of memory, improved diagnosis functions, and an enhancement to the SAP GUI for HTML for Mozilla and Firefox. For more information, see SAP Note 890606: *SAP NetWeaver 2004s integrated ITS: New Features*.

Exercise 2: Administrative Work with the ICF

Exercise Objectives

After completing this exercise, you will be able to:

- Activate and call ICF services
- Create external aliases
- Use the IFC recorder
- Activate and call the integrated ITS

Business Example

As part of SAP ERP, your company implements the SAP ECC 6.0 software component. Your development department has added a number of BSP-based applications to the standard SAP system. You are to ensure that these applications can be called.

Task 1: Activating ICF Services

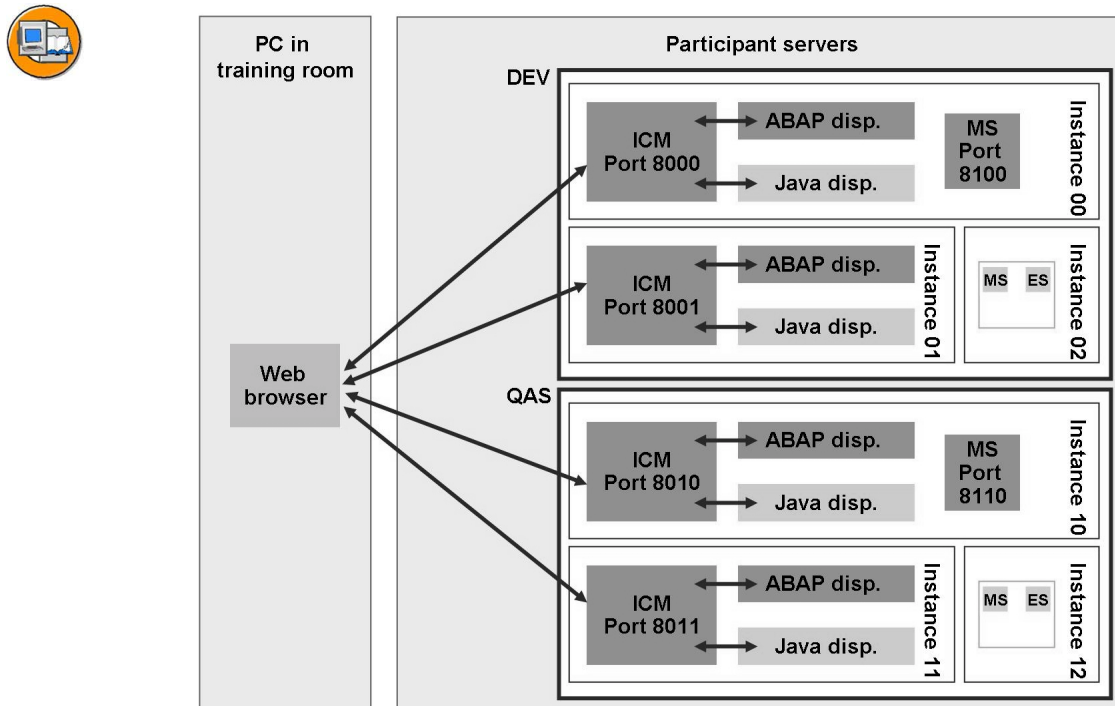


Figure 22: Complete Scenario of the Training Landscape

Activating and Calling ICF Services

1. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/bc/bsp/sap/it00`**
(example for group QAS and server twdf0042:
`http://twdf0042.wdf.sap.corp:8010/sap/bc/bsp/sap/it00`).
- Why do you get an error message?
2. Activate the ICF service `/sap/bc`, including all subservices.
3. Enter the URL **`http://<server with domain>:<ICM port>/sap/bc/bsp/sap/it00`**.
4. Within BSP application `it00`, start the function *MIMEs* → *Accessing SAP Icons in MIME Repository*. Note the Internet server cache (of the ICM that processed your request).

Continued on next page

5. **Optional:** From transaction SICF, start the Web dynpro ABAP application `/sap/bc/webdynpro/sap/OTHELLO`.

Result

All services under `/sap/bc` can now be called from the intranet and, if the network settings allow, from the Internet.

Task 2: Creating an External Alias

Create a new ICF service as an external alias to an existing service.

1. Use the wizard to create a new ICF service `/ADM102/myInfo` that links to `/sap/bc/icf/info` as an external alias.
2. Close all of your browser windows. What do you notice when you test the new service `/ADM102/myInfo` in transaction SICF?
3. Enter the logon data for your **ADM102-##** user in the `/ADM102/myInfo` service.
4. Check that you can now call the `/ADM102/myInfo` service without having to log on.

Result

Your SAP system provides a new `/ADM102/myInfo` service with saved logon data.

Task 3: Optional: Creating an Internal Reference

Create a new service as an internal reference to an existing service.

1. Without using the wizard, create an ICF service `/ADM102/myPing` that links to `/sap/public/ping` as an internal reference. Note that you first require an independent service `ADM102` below which you enter `myPing` as an internal reference.

Remember to activate your new service.

2. Test the new `/ADM102/myPing` service in transaction SICF.
3. **Optional:** Determine all ICF services that link to the `/sap/public/ping` service.

Result

Your SAP system provides a new service, `/ADM102/myPing`.

Continued on next page

Task 4: ICF Recorder

Analysis of HTTP requests with the ICF recorder

1. Activate the ICF recorder for service `/sap/bc/bsp/sap/it00`. The recording should contain requests, responses, and failed logons, be relevant to all users, and be stored for one day.
2. Close all open Web browser windows.

Start the `/sap/bc/bsp/sap/it00` service (either by using the test function in transaction SICF or by entering the URL in the Web browser) and log on with...

... an invalid user name.

... a valid user name but an incorrect password.

... a valid user name and password.

Once you have logged on successfully, call some of the functions.
3. Deactivate the ICF recorder
4. Display the recorded requests and responses.

Task 5: SAP GUI for HTML Using Integrated ITS

Set up and call the SAP GUI for HTML for the integrated ITS in AS ABAP.

1. Check whether the integrated ITS is active.
2. Verify that the ICF service `/sap/bc/gui/sap/its/webgui` is active and intended for calling the GUI.

Verify that the ICF service `/sap/public/bc/its/mimes` is active and that *Not Specified* is entered for the GUI link.
3. Start the SAP GUI for HTML from transaction SICF.

Can you start the transaction from the command field?
4. **Optional:** Change the service parameter `~noHeaderOkCode` for the SAP GUI for HTML to **1** and note its effects.

Result

Users can log onto the SAP system with a Web browser and call transactions with the SAP GUI for HTML.

Solution 2: Administrative Work with the ICF

Task 1: Activating ICF Services

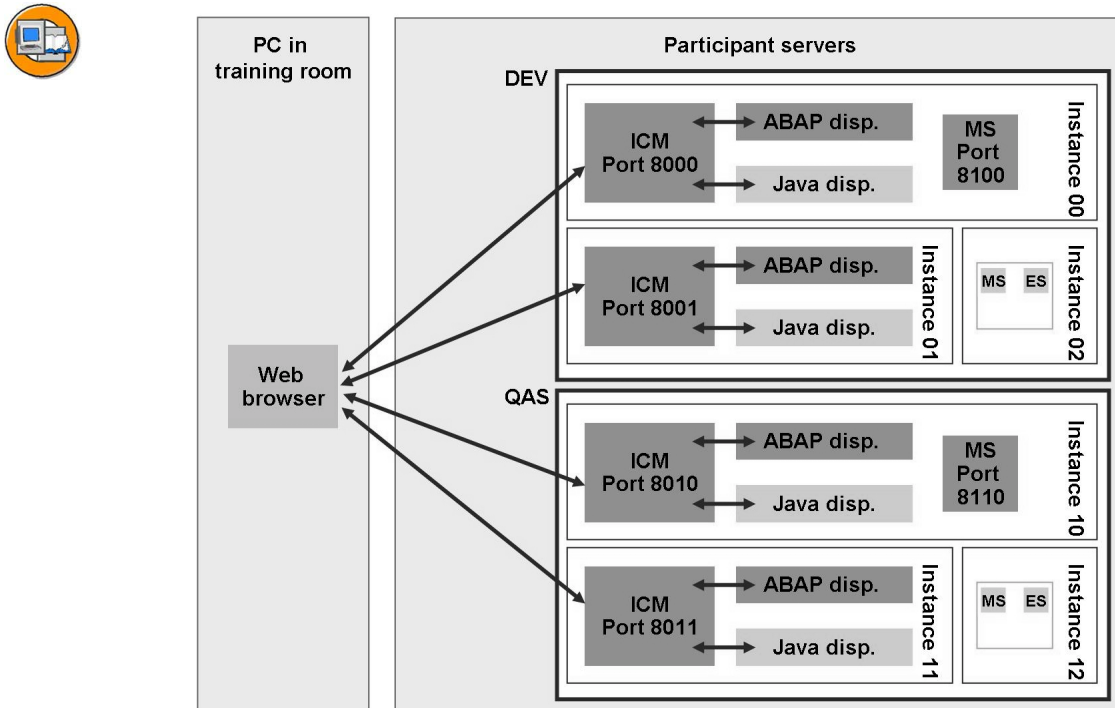


Figure 23: Complete Scenario of the Training Landscape

Activating and Calling ICF Services


1. Call the following URL in your local Web Browser (in the training room): **`http://<server with domain>:<ICM port>/sap/bc/bsp/sap/it00`**
(example for group QAS and server twdf0042:
`http://twdf0042.wdf.sap.corp:8010/sap/bc/bsp/sap/it00`).

Why do you get an error message?

- a) Enter the specified URL in your local Web browser and choose *Enter*.
- b) The error message “Service cannot be reached” appears since the service is inactive (this is the initial status of all services provided by SAP).

Continued on next page

2. Activate the ICF service */sap/bc*, including all subservices.
 - a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
 - b) Navigate to the */sap/bc* node.
 - c) Click this node with the secondary mouse button and choose *Activate Service* (or choose *Service/Host* → *Activate*).

Choose *Yes* (with the tree icon) to activate node */sap/bc* and all of its subservices.
3. Enter the URL **http://<server with domain>:<ICM port>/sap/bc/bsp/sap/it00**.
 - a) Enter the specified URL in your local Web browser again and choose *Enter*.
 - b) Log on with your **ADM102-##** user.
 **Note:** BSP application *it00* is a test application for BSP developers.
4. Within BSP application *it00*, start the function *MIMEs* → *Accessing SAP Icons in MIME Repository*. Note the Internet server cache (of the ICM that processed your request).
 - a) Within BSP application *it00* (in the Web browser), navigate to the menu option *MIMEs* → *Accessing SAP Icons in MIME Repository*.

The system displays the SAP icons stored in the MIME repository (implemented in the AS ABAP database) as graphics in GIF format.
 - b) Call transaction SMICM on the application server whose ICM processed the previous request.
 - c) Choose *Goto* → *HTTP Server Cache* → *Display*.

The system displays the objects held in the ISC (with a preview option). If you now delete the temporary cache in your Web browser (in Internet Explorer choose *Tools* → *Internet Options* → *General* → *Temporary Internet Files* → *Delete Files*) and call the function *MIMEs* → *Accessing SAP Icons in MIME Repository* again, you can clearly see the difference between the read from the database and the read from the ISC.

Continued on next page

5. **Optional:** From transaction SICF, start the Web dynpro ABAP application `/sap/bc/webdynpro/sap/OTHELLO`.
 - a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
 - b) Navigate to the `/sap/bc/webdynpro/sap/OTHELLO` node.
 - c) Click the node with the secondary mouse button and choose *Test Service*. When prompted, log on with your **ADM102-##** user. Good luck...

Result


All services under `/sap/bc` can now be called from the intranet and, if the network settings allow, from the Internet.

Task 2: Creating an External Alias


Create a new ICF service as an external alias to an existing service.

1. Use the wizard to create a new ICF service `/ADM102/myInfo` that links to `/sap/bc/icf/info` as an external alias.
 - a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
 - b) Choose *External Aliases* to switch to the *Maintain External Aliases* screen.
 - c) Select *default_host*.
 - d) Choose *Wizard – External Alias* (or *External Alias* → *Wizard – External Alias*).
 - e) The wizard leads you through the following steps, whereby you access the following page by choosing *Continue*:
 - First, information about external aliases appears.
 - In the *External Alias* field, enter `/ADM102/myInfo` and in the *Description* field, enter a description of your choice.
 - As the target handler, double-click the service `/sap/bc/icf/info`.
 - Choose *Complete*.

Continued on next page

2. Close all of your browser windows. What do you notice when you test the new service */ADM102/myInfo* in transaction SICF?
 - a) In the hierarchy display in transaction SICF (*Maintain External Alias* view), click your new node */ADM102/myInfo* with the secondary mouse button and choose *Test Ext. Alias*.
 - b) Note: You must first log on (with your **ADM102-##** user) before you can view the XML document with the system information.
 **Note:** If you are not prompted to enter your user details, close all of your browser windows.
3. Enter the logon data for your **ADM102-##** user in the */ADM102/myInfo* service.
 - a) In the hierarchy display in transaction SICF (*Maintain External Alias* view), double-click your */ADM102/myInfo* node and switch to change mode.
 - b) On the *Logon Data* tab page, select *Required with Logon Data* from the *Procedure* dropdown box.

In the *Logon Data* area, enter the data for your **ADM102-##** user. Confirm the warning “If possible, do not use dialog users” and save your entries.

 **Note:** In an operational scenario, enter service users with relevant authorizations.
4. Check that you can now call the */ADM102/myInfo* service without having to log on.
 - a) Close all of your browser windows.
 - b) In transaction SICF, click your */ADM102/myInfo* external alias with the secondary mouse button and again choose *Test Ext. Alias*. The system should not prompt you to enter your user details.

Result

Your SAP system provides a new */ADM102/myInfo* service with saved logon data.

Continued on next page

Task 3: Optional: Creating an Internal Reference

Create a new service as an internal reference to an existing service.

1. Without using the wizard, create an ICF service `/ADM102/myPing` that links to `/sap/public/ping` as an internal reference. Note that you first require an independent service `ADM102` below which you enter `myPing` as an internal reference.

Remember to activate your new service.

- a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
- b) In the hierarchy display, click the top node `default_host` with the secondary mouse button. Choose *New Sub-Element* (or *Service/Virtual Host* → *Create Service*).

Confirm the message about the namespace concept.

- c) In the *Create a Service Element* window, enter **ADM102** as the name and *Independent Service* as the *Type*.
- d) On the subsequent *Create/Change a Service* screen, enter a description. Leave all other fields unchanged and save your changes (repository request: *Local Object*). Confirm the warning that the new service cannot be accessed. Go back one screen (to the hierarchy display in transaction SICF).
- e) In the service tree, select your new `ADM102` node, click the secondary mouse button, and choose *New Sub-Element*. Now create the element **myPing** as a *Reference to Existing Service*.
- f) On the *Create/Change a Service Call* screen, enter a description text.

On the *Alias Trgt* tab page, navigate to the `default_host/sap/public/ping` node and double-click this line.

Save your data and go back one screen (to the hierarchy display in transaction SICF).

- g) To activate your changes, select the `/ADM102/myPing` node in the service tree and choose *Activate Link* with the secondary mouse button. Confirm the query by choosing *Yes*.
2. Test the new `/ADM102/myPing` service in transaction SICF.
 - a) In the hierarchy display in transaction SICF, click your new `/ADM102/myPing` node with the secondary mouse button and choose *Test Link*. The message “Server reached successfully” appears.

Continued on next page

3. **Optional:** Determine all ICF services that link to the `/sap/public/ping` service.

- a) In transaction SICF, navigate to the node `/sap/public/ping`. With the secondary mouse button, choose *References to Service* (or choose *Service/Host* → *Alias References*).

The system displays a list that contains all internal and external references that link to `/sap/public/ping`. Your entry, `/ADM102/myPing`, also appears.

Result

Your SAP system provides a new service, `/ADM102/myPing`.

Task 4: ICF Recorder

Analysis of HTTP requests with the ICF recorder

1. Activate the ICF recorder for service `/sap/bc/bsp/sap/it00`. The recording should contain requests, responses, and failed logons, be relevant to all users, and be stored for one day.

- a) In the hierarchy display in transaction SICF, select the `/sap/bc/bsp/sap/it00` service with the primary mouse button and choose *Edit* → *Recorder* → *Activate Recording*.
- b) In the *Activate Recording* dialog box, deselect the *User-Dependent* field. In the *Lifetime* field, enter **1** day and **00:00:00** hours:minutes:seconds. Under *Recording Level*, select the *Logon* field and the option *Request+Response*, and then choose *Activate*.
- c) The red flag on the *System Monitor Active* pushbutton indicates the changed operation mode. If you choose this pushbutton, the *Monitor Current System Settings* dialog box appears with details.

2. Close all open Web browser windows.

Start the `/sap/bc/bsp/sap/it00` service (either by using the test function in transaction SICF or by entering the URL in the Web browser) and log on with...

... an invalid user name.

... a valid user name but an incorrect password.

... a valid user name and password.

Once you have logged on successfully, call some of the functions.

- a) See the task.

Continued on next page

3. Deactivate the ICF recorder

- a) To deactivate the recording for */sap/bc/bsp/sap/it00* only, first select this service with the primary mouse button in transaction SICF. Then choose *Edit → Recorder → Deactivate Recording*.
- b) Accept the entries in the *Deactivate Recording* dialog box and choose *Deactivate*.

A list appears containing the active recordings for the selected server. Choose *Deactivate* again.

- c) Provided that no other recordings are active in the system, the pushbutton for the system status displays a green flag and the text *System Monitor Inactive*.



Hint: To determine the services for which a recording is active, simply choose *Whole Application Server* in the *Deactivate Recording* dialog box. The list that is displayed (that you can also cancel) contains a column with the monitored URLs.

4. Display the recorded requests and responses.

- a) Call transaction SICF and choose *Edit → Recorder → Display Recording*.



Note: Alternatively, call transaction SICFRECORDER.

- b) To display an overview of all recordings, delete any entries in the *Request Path* field on the selection screen. Select *Logon Error*. Accept the remaining entries and choose *Execute*.
- c) On the left of the screen, the system lists the services for which there are recordings. If you double-click an ICF service, the details of the requests and responses are displayed on the right of the screen.

Continued on next page

Task 5: SAP GUI for HTML Using Integrated ITS

Set up and call the SAP GUI for HTML for the integrated ITS in AS ABAP.

1. Check whether the integrated ITS is active.
 - a) Check that profile parameter *itsp/enable* is set to **1**, either by:
 - Executing report *RSPFPAR*
 - Executing report *SITSPMON*
 - Calling transaction RZ11Note that all of these approaches are instance-specific.
2. Verify that the ICF service */sap/bc/gui/sap/its/webgui* is active and intended for calling the GUI.

Verify that the ICF service */sap/public/bc/its/mimes* is active and that *Not Specified* is entered for the GUI link.

 - a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
 - b) Navigate to the service */sap/bc/gui/sap/its/webgui* and activate it if necessary (click the secondary mouse button and choose *Activate Service*). Double-click the service and make sure that the *GUI Link* field on the *Service Data* tab page contains the value *Yes*.

Navigate to the service */sap/public/bc/its/mimes* and activate it if necessary (click the secondary mouse button and choose *Activate Service*). Double-click the service and make sure that the *GUI Link* field on the *Service Data* tab page contains the value *Not Specified*.
3. Start the SAP GUI for HTML from transaction SICF.

Can you start the transaction from the command field?

 - a) In the hierarchy tree in transaction SICF, click the */sap/bc/gui/sap/its/webgui* node with the secondary mouse button and choose *Test Service*.
 - b) Log onto the SAP GUI for HTML Web browser with your **ADM102-##** user and password. The system may display warnings about the logon procedure. If so, choose *Log On* and enter your details.

Choose the small triangle in the top-left of the screen to display the command field in which you can enter transaction codes.

Continued on next page

4. **Optional:** Change the service parameter *~noHeaderOkCode* for the SAP GUI for HTML to **1** and note its effects.
 - a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
 - b) Navigate to the service */sap/bc/gui/sap/its/webgui* and double-click it. Switch to change mode. On the *Service Data* tab page, choose *GUI Configuration*.
 - c) In the *Maintain Service Parameters* dialog box, enter **~noHeaderOkCode** in an empty row in the *Parameter Name* column, and enter **1** in the *Value* column.

Confirm your entries by choosing *Copy Parameter Set*.
 - d) On the *Create/Change a Service* screen, choose *Save*. If necessary, create a new workbench request and leave the screen by choosing *Back*.
 - e) Start the SAP GUI for HTML from transaction SICF (as previously described).
 - f) Log onto the Web browser. You now cannot navigate in the SAP system using the command field.

Result

Users can log onto the SAP system with a Web browser and call transactions with the SAP GUI for HTML.



Lesson Summary

You should now be able to:

- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Activate and use the integrated ITS as of AS ABAP 6.40

Related Information

- Online documentation for SAP NetWeaver 7.0 under *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *Platform-Wide Services* → *Connectivity* → *Components of SAP Communication Technology* → *Communication Between ABAP and Non-ABAP Technologies* → *Internet Communication Framework*
- Online documentation for SAP NetWeaver 7.0 under *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *ABAP Technology* → *UI Technology* → *Web UI Technology* → *ITS/SAP@Web Studio* → *SAP ITS in the SAP Web Application Server*.
- SAP Note 517484: *Inactive services in the ICF*
- SAP Note 685575: *Corrections to SAP Web AS 6.40 in the ICF*
- SAP Note 709038: *SAP Integrated ITS*
- SAP Note 721993: *SAP ITS Release 6.40: Corrections*
- SAP Note 890601: *SAP ITS Release 7.00: Corrections*
- SAP Note 698329: *Integrated ITS, WEBGUI/IAC logon fails*
- SAP Note 890606: *Integrated ITS: New features for 7.00*
- SAP Note 732218: *ICF: Logon data from SICF is not transported.*
- Course **NET200** *SAP Web AS: Developing BSP Applications*

Lesson: The SAP Web Dispatcher

Lesson Overview

This lesson highlights what the SAP Web Dispatcher is used for and how to operate it.



Lesson Objectives

After completing this lesson, you will be able to:

- Outline the function of the SAP Web Dispatcher
- Explain how you can use the SAP Web Dispatcher to distribute workload across the different instances of an SAP system

Business Example

Your company offers its customers browser-based access to data from SAP systems. Technically, the Web applications in ABAP and Java are implemented on the SAP NetWeaver AS. Since a very large number of your customers use this service, your SAP system, which is connected to the Internet, has multiple instances. It is your task as a member of the system administration team to implement load distribution across the different instances of this SAP system.

Implementation Area of the SAP Web Dispatcher

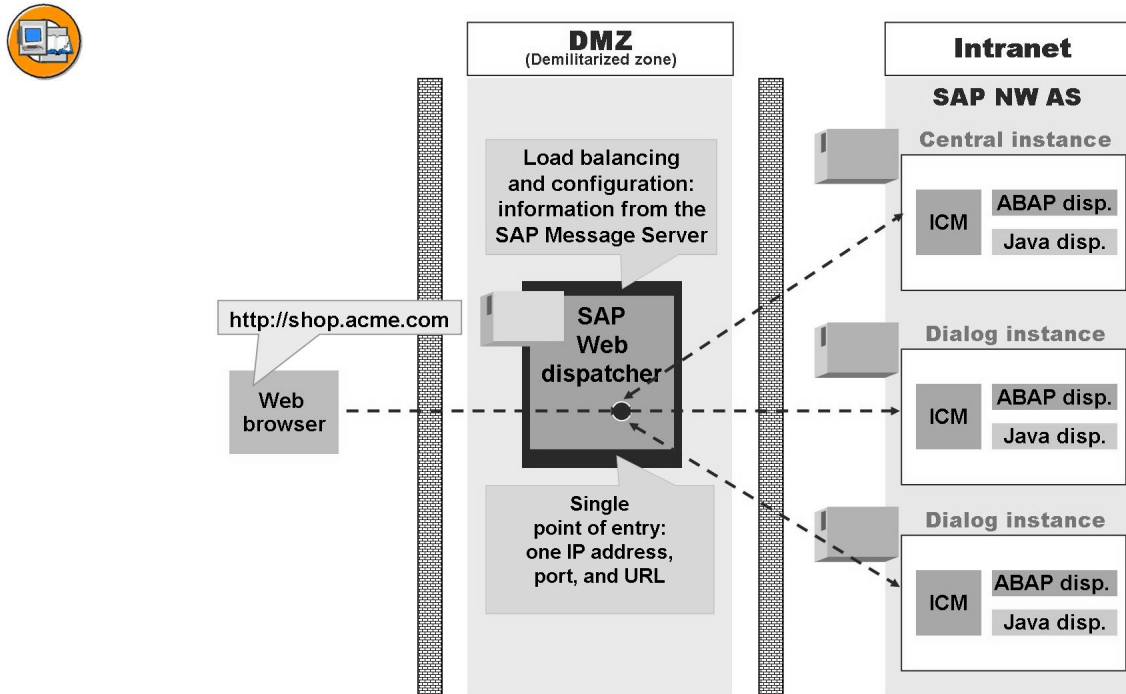


Figure 24: An overview of the SAP Web Dispatcher

You should always consider using the SAP Web Dispatcher if the Web applications that you are operating in your SAP system are implemented as applications in ABAP (for example, BSPs) and/or in Java/J2EE.

Some of the requirements that motivated the development of the SAP Web Dispatcher:

- The Web applications are also to be used from the Internet. The company network is protected by a Demilitarized Zone (DMZ) and the critical business processes run on servers that are not recognized on the Internet. How can you avoid the need to place an SAP application server within the DMZ?
- The SAP system in question consists of multiple application servers (instances) that are distributed across multiple hosts. However, the Web applications provided should run under a descriptive address; technical details such as server name and port number are to remain hidden to users.
- What form can a sensible load distribution take? All application servers may not provide all services (ICF or J2EE Engine).

Requirements such as these can be implemented using third-party products known as reverse proxies or Web switches. Although there are advantages, such as high throughput and implementation in close proximity to the hardware, these must be offset against the disadvantages of additional costs and restricted SAP integration.

The SAP Web Dispatcher, delivered as of SAP Web AS 6.20, acts like a “software Web switch”. It is a stand-alone program that you can run on a separate host without any additional software. In this way, the SAP Web Dispatcher implements a central entry point for HTTP(S) requests to an SAP system, including load distribution across multiple instances.



Hint: The SAP Web Dispatcher is only relevant in the Web environment. When a SAP GUI for Windows or SAP GUI for Java is used for access, the ABAP message server distributes the load.



Hint: In principle, the message server can also carry out a load distribution for HTTP-based requests, but its functions are restricted in contrast to the SAP Web Dispatcher, which is why SAP recommends that you use the SAP Web Dispatcher. For more information, see SAP Note 1040325: *HTTP load balancing: Message Server or Web Dispatcher?* In particular, the message server should not be used to distribute loads for the CRM system, for example, if requests are made to an SAP CRM system from an SAP NetWeaver Portal.

Functions of the SAP Web Dispatcher

The SAP Web Dispatcher ultimately forwards an HTTP(S) request to a specific application server. This section outlines the criteria by which this is performed.

An HTTP request (or unpacked HTTPS request) is assigned to a server in two stages:

1. First, the SAP Web Dispatcher determines whether the incoming HTTP request is to be forwarded to an ABAP or Java server. It then finds a group of servers in the SAP system that can execute the request.
2. Load balancing is then carried out within this group. After the SAP Web Dispatcher has identified a server, it forwards the request to the ICM of the relevant application server.

➔ **Note:** A SAP Web Dispatcher can distribute requests for only **one** SAP system. If multiple SAP systems are required, you have to set up and start separate SAP Web Dispatcher processes for each of the respective systems (which can run together on one computer).

The SAP Web Dispatcher is backwards compatible, that is, the SAP Web Dispatcher release can be higher or the same as the back-end system release. The patch level can also differ from the patch level of the back-end system. For back-end systems based on SAP Web AS 6.40 or SAP NetWeaver AS 7.00, we recommend using SAP Web Dispatcher 7.00 provided that this is compatible with the operating system used. For more information, see SAP Note 538405 - *Composite SAP Note: SAP Web Dispatcher*.

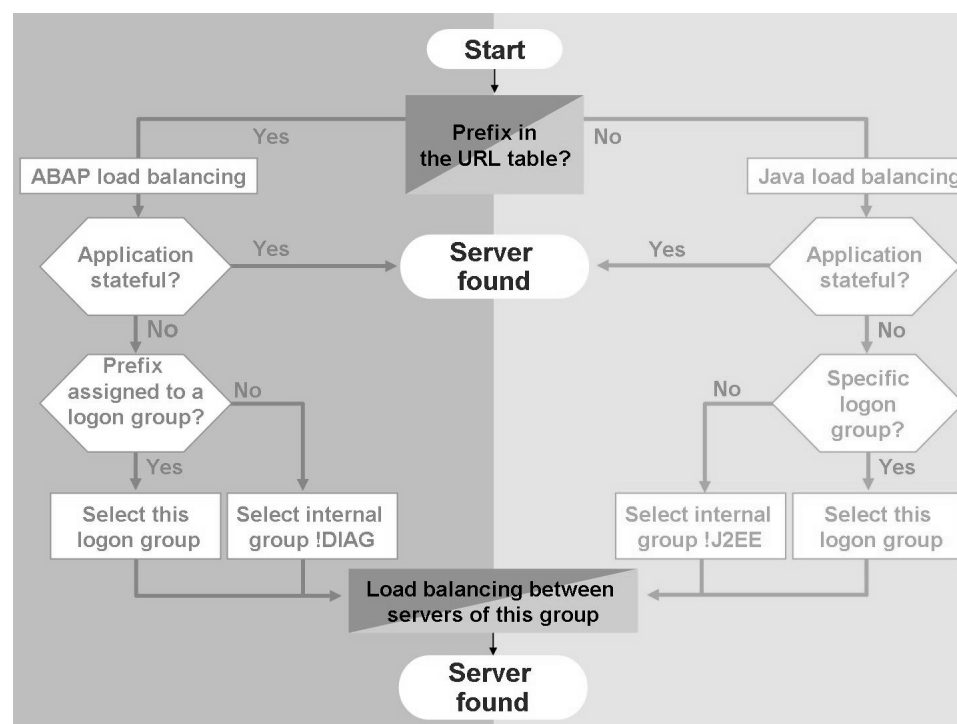


Figure 25: From the HTTP(S) Request to the Application Server (Simplified)

Server Selection

The SAP Web Dispatcher first checks whether the request is an ABAP or J2EE request. This distinction is based on the analysis of the URL prefix. For the URL *http://adm102.sap.com/A/B/C/Default.html*, the prefix to be analyzed is the character string */A/B/C/*. If this prefix is known in the ICF, this is an ABAP request. If the URL contains only **one** forward slash (/) after the host name, special handling is required: The value of the profile parameter *is/HTTP/default_root_hdl* determines the destination.

In the case of an **ABAP request**, the SAP Web Dispatcher first uses a cookie to identify whether the request concerns a stateful application. If this exists, the decision is simple. The request is sent to the application server that is processing this session. For a stateless application, the internal group *!DIAG* is selected, which consists of all ABAP application servers. This is used only if a logon group (maintained with transaction *SMLG*) is inherited or specified explicitly in the ICF service.

AS Java also recognizes the concept of logon groups. If a specific logon group has not been configured for the prefix of the **Java request** called, the SAP Web Dispatcher uses the internal group *!J2EE*. In the case of a stateful application, this is indicated through the session information in the URL or a load-balancing cookie. For compatibility reasons, the session cookie *jsessionid* can also still be used here.

The SAP Web Dispatcher obtains information about the logon groups and URL mapping from an ABAP application server via HTTP or HTTPS. For this to happen, the services */sap/public/icman* and */sap/public/icf_info/** must be activated in the ICF.

Load Balancing

The SAP Web Dispatcher obtains information about the application servers of the SAP system from the message server via HTTP(S). You can use the SAP Web Dispatcher in pure ABAP systems as well as in combined ABAP + Java systems and pure Java systems. In this way, the installation option determines the message server with which the SAP Web Dispatcher communicates.

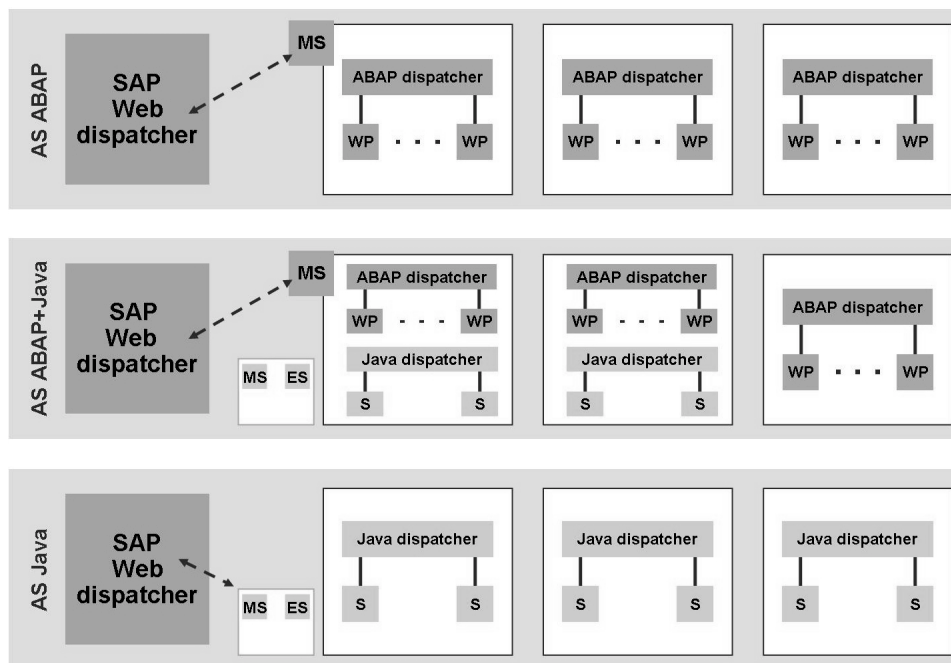


Figure 26: Communication Partners of the SAP Web Dispatcher



Communication Partners of the SAP Web Dispatcher for Server Information

SAP Web Dispatcher...	Installation option SAP NetWeaver AS		
	ABAP	ABAP + Java	Java
...communicates with the	ABAP message server		Java message server
...parameters <i>rdisp/mshost</i> and <i>ms/http_port</i> correspond with	the host name of the message server as well as its parameter <i>ms/server_port_<xx></i>		



Note: The **SAP Web Dispatcher** uses the parameters *rdisp/mshost* and *ms/http_port* to specify the assigned message server of the SAP system.

For compatibility reasons, the obsolete parameters *ms/http_port* and *ms/https_port* still function in the SAP system for the **ABAP message server** but should no longer be used.

The HTTP interface of the message server allows you to display information about the application server with a Web browser. To do so, call the URL `http://<message server with domain>:<message server port>/msgserver/commands`.


The SAP Web Dispatcher distributes the requests in turn within the selected server group by default, weighted according to the capacity of the individual servers. For ABAP, the capacity is calculated from the number of dialog work processes and, for Java, from the number of server processes. You can use the profile parameter *wdisp/load_balancing_strategy* (explained in detail in the online documentation) to configure the SAP Web Dispatcher for different load-balancing procedures.

Operating the SAP Web Dispatcher

The internal structure of the SAP Web Dispatcher is based on the ICM process. A profile file is also used in this case to determine the settings with which the SAP Web Dispatcher is started. You can easily copy and use the executable file (*sapwebdisp.exe*) that SAP makes available for all supported operating systems to a separate host, together with the profile.

Profile Parameters

For templates for the profile and parameter descriptions, see the online documentation. The SAP Web Dispatcher essentially only needs to know the port at which it is to receive HTTP requests (parameter *icm/server_port_<x>*) and on which host (*rdisp/mshost*) and with which HTTP port (*ms/http_port*) it can access the message server.

 **Note:** If metadata is to be exchanged via HTTPS, additional steps are required (see online documentation).

As of SAP Web AS 6.40, you can also start the SAP Web Dispatcher without a profile file. For this bootstrap option (started with command **sapwebdisp -bootstrap**), the following steps are carried out:

1. If the profile file *sapwebdisp.pfl* does not exist already, it is created based on interactive entries.
2. If the authorization file *icmauth.txt* does not exist, it is created and a user is entered for Web administration (see below).
3. The SAP Web Dispatcher is started with the profile file created.

Go Live (Without Bootstrap Option)

You start the SAP Web Dispatcher with the operating system command **sapwebdisp pf=<profile file>**, where you can set additional options such as trace file and trace level (see the online documentation).



Hint: In Microsoft Windows, you can set up the SAP Web Dispatcher as a service with the command **ntscmgr install sapwebdisp -b <program path>\sapwebdisp.exe -p "service pf=<profile file> <options>"**.

You stop the SAP Web Dispatcher using a **kill** command at operating system level. To do this, you require its process ID (PID), which you can identify from the output when it is started or from the trace file (by default *dev_sapwebdisp*). Under Unix, the command is **kill -2 <PID>**; under Microsoft Windows, it is **sapntkill -INT <PID>**.


Monitoring

You can monitor the SAP Web Dispatcher with the *icmon* command line program (introduced in the ICM lesson). When doing so, do not change the profile file for the SAP Web Dispatcher.

As of SAP Web AS 6.40, a **Web-based interface** is available for SAP Web Dispatcher administration and monitoring. To use this, the following prerequisites must be met:

- You have unpacked the SAP Web Dispatcher installation package in a directory.
- The *icmauth.txt* file exists for authorizing administrators.
- In the SAP Web Dispatcher profile, the *icm/HTTP/admin_<xx>* parameter is maintained, which you can use to further restrict access to the Web administration interface (for example, to specific ports or host names). When the SAP Web Dispatcher is started with the bootstrap option, a default value is generated for this parameter automatically.





Admin

http://<server with domain>:<SAP Web Dispatcher port>/<Admin prefix>

SAP Web Dispatcher Menu

- Core System
 - Monitor
 - Active Services
 - Active Connections
 - Trace
 - Statistics
 - Instance Buffer
 - Release Information
 - Static
 - MP Status
 - ICM Security Log
- HTTP Handler
 - Access Log
 - Session Cache
 - Access Handler
 - Admin Handler
- Dispatching Module
 - Monitor Server Groups
 - SSL End To End Dispatching
 - URL Mapping
 - URL Error
 - Statistics
 - Session Dispatching

Welcome, franky!
Admin rights granted

Server Group Monitor Detailed Web Dispatcher configuration

powered by **ICM**

State of Server Group "123E"

Loadbalancing Information

Number of Servers in this group: 2

Last used Server: tw01123_wd1 sap.com

Preferred next Server: tw01123_OAS_11

Active Hosts									
Nr.	Valid	Name	Hostname	HTTP conn pool	HTTPS conn pool	Capacity	Requests	Load	Response time (ms)
0	✓	tw01123_OAS_11	tw01123_wd1 sap.com	Port max 100 / max 100	Port max 100 / max 100	51101	1	1000	0 Aug 2006 00:00
1	✓	tw01123_OAS_10	tw01123_wd1 sap.com	Port max 100 / max 100	Port max 100 / max 100	51101	1	1000	0 Aug 2006 00:00

- SAP Web dispatcher monitoring
- Display parameter settings
- Statistics
- Evaluation of trace files
- User administration

Figure 27: Functions of the Web Admin Interface

Then call the Web admin interface using the URL **http://<server with domain>:<SAP Web Dispatcher port>/<Admin prefix>**. The **<Admin prefix>** section is defined through the assignment for *PREFIX* of the profile parameter *icm/HTTP/admin_<xx>*. The value */sap/wdisp/admin* is the default value for the bootstrap option.


Note: For security reasons, use the HTTPS log for administration. If you use HTTP, administrator passwords are transferred without encryption and can be tapped.

Your logon data (user and password) is checked against the *icmauth.txt* file. If you start the SAP Web Dispatcher with the bootstrap option, this is generated automatically. You edit the entries in the *icmauth.txt* authentication file using the *icmon* program by calling **icmon -a**.

Once you have logged on successfully, the administration and monitoring interface is displayed, which is divided into a navigation area (left side) and a detail area (right side).

2008

© 2008 SAP AG. All rights reserved.

79 

Exercise 3: Administration of the SAP Web Dispatcher

Exercise Objectives

After completing this exercise, you will be able to:

- Configure the SAP Web Dispatcher
- Start and stop the SAP Web dispatcher
- Use and monitor the SAP Web dispatcher

Business Example

Your company is planning to go live with an extensive Web application based on SAP Web AS. Due to the expected load and as a fail-safe, you are planning to use multiple application servers on different hosts. You are responsible for ensuring that the applications can be called on the Internet using a simple URL.

Task 1: Configuring the SAP Web Dispatcher

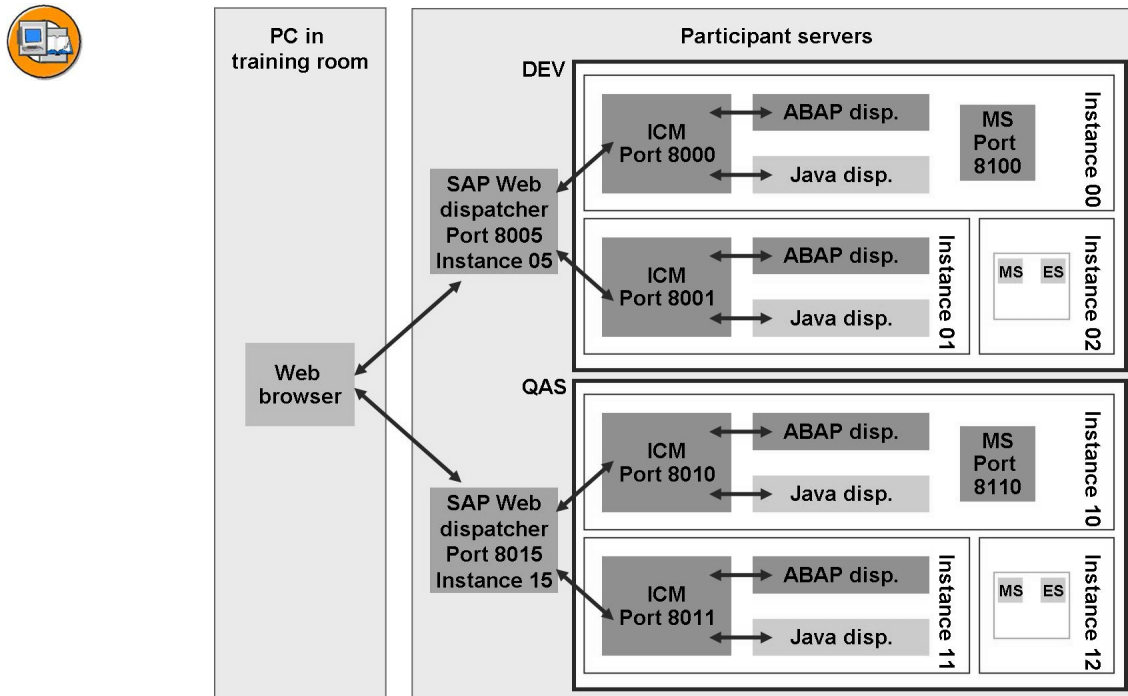


Figure 28: Complete Scenario of the Training Landscape

Determine the release of the SAP Web Dispatcher, and use the bootstrap option to configure and start it.

➔ **Note:** In practice, the SAP Web dispatcher would run on a separate host (possibly with Internet access) under a descriptive name. On the training systems for the ADM102 course, the SAP Web Dispatcher and the SAP system with two instances run on one host.

1. Check that the ICF services under `/sap/public/icf_info` are active.
Test the ICF service `/sap/public/icf_info/urlprefix`.
2. As part of the setup, determine the HTTP port of the ABAP message server for your SAP system.
3. On your server, create directory `E:\SAPWebDisp##`, where ## is your group number.

➔ **Note:** Do not use spaces in the directory name.

Continued on next page

4. Copy the installation package of the SAP Web Dispatcher from the training share to the new directory.
5. Unpack the installation package of the SAP Web Dispatcher.
6. Which release of the SAP Web Dispatcher is used on the training system?
7. Start the SAP Web Dispatcher with the bootstrap option and configure it for your SAP system. Use the following entries:

	DEV Group	QAS Group
<i>Hostname of Message Server</i>	Full host name of your server, for example twdf0042.wdf.sap.corp	
<i>HTTP Port of Message Server</i>	8100	8110
<i>Unique Instance Number for SAP Web Dispatcher</i>	05	15
<i>HTTP port number for SAP Web Dispatcher</i>	8005	8015
<i>Create configuration for ...</i>	s(mall) system	



Caution: Remember to note the password that is generated for the *icmadm* user.

8. Open the profile file of the SAP Web Dispatcher that was created, and note how your entries have been implemented. What is the URL used to call the Web admin interface?

Result

The SAP Web Dispatcher is configured and has started for your SAP system.

Task 2: Using the SAP Web Dispatcher

Send requests that are distributed using the SAP Web dispatcher.

1. Call the ICF service `/ADM102/myInfo` using your SAP Web Dispatcher.



Note: If you have not completed the relevant task in the “Internet Communication Framework” lesson, use the ICF service `/sap/bc/icf/info` instead.

Continued on next page

2. Create a *Dialog* logon group that includes only the ABAP dialog instance. Define this logon group for the external alias `/ADM102/myInfo`.

Note what happens when you call the ICF service `/ADM102/myInfo` using your SAP Web Dispatcher.
3. Call the Web dynpro Java application `/webdynpro/dispatcher/local/WhoAmI/Show` using your SAP Web Dispatcher.

Result

You have learned about the SAP Web Dispatcher's mode of operation.

Task 3: Restart the SAP Web Dispatcher Using Operating System Tools

Start and stop the SAP Web Dispatcher using operating system tools.

1. Stop your SAP Web Dispatcher.
2. Start the SAP Web Dispatcher with the generated profile file.

Result

You can start and stop the SAP Web Dispatcher at operating system level.

Task 4: Optional: Set Up as Windows Service

Set up the SAP Web Dispatcher as a Windows service that is started automatically.

1. Stop the SAP Web dispatcher using operating system tools.
2. Configure your SAP Web Dispatcher as Windows service `SAPWebDisp##` and start it. This service should start automatically when you boot the host.

Result

You can configure the SAP Web Dispatcher under Windows as a service.

Task 5: Web Admin Interface

Use the Web-based interface for administration and monitoring of the SAP Web Dispatcher.

1. Call the Web admin interface using a Web browser.
2. Change the password for the *icmadm* user.
3. Determine which servers are available for which logon groups.

Continued on next page

4. **Optional:** Use the *icmon* program to create another administration user **WebDispADM##** with a password of your choice.

Result

You can now configure and use the easy Web admin interface of the SAP Web Dispatcher.

Solution 3: Administration of the SAP Web Dispatcher

Task 1: Configuring the SAP Web Dispatcher

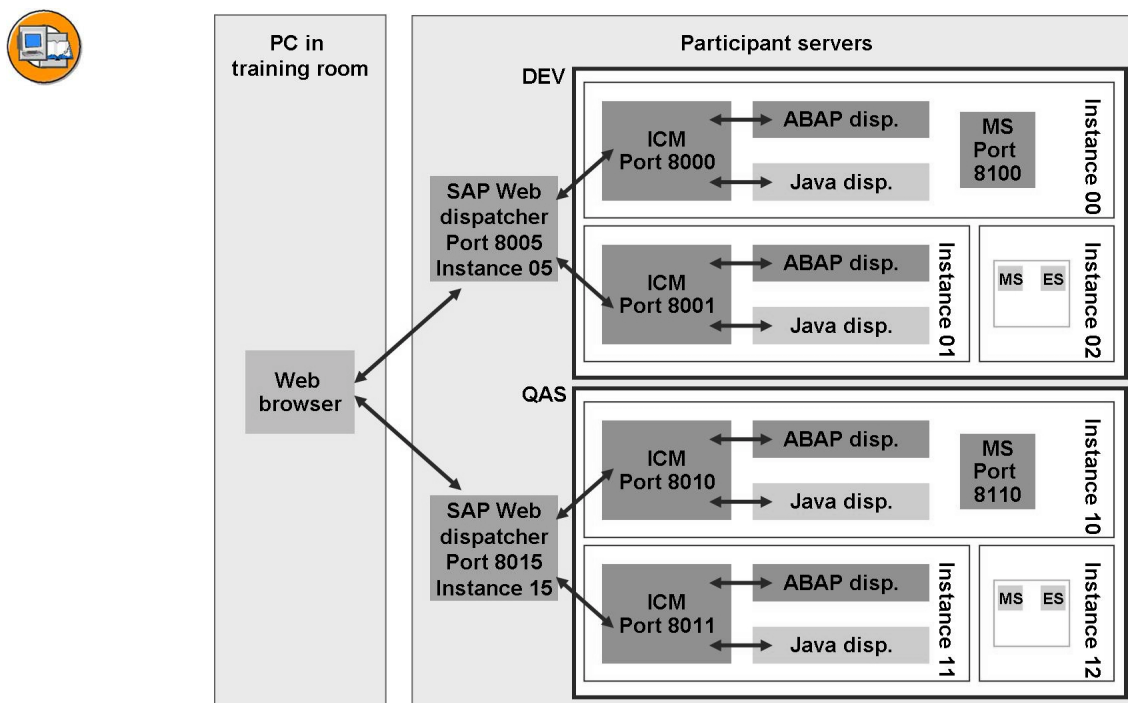


Figure 29: Complete Scenario of the Training Landscape

Determine the release of the SAP Web Dispatcher, and use the bootstrap option to configure and start it.

➔ **Note:** In practice, the SAP Web dispatcher would run on a separate host (possibly with Internet access) under a descriptive name. On the training systems for the ADM102 course, the SAP Web Dispatcher and the SAP system with two instances run on one host.

1. Check that the ICF services under `/sap/public/icf_info` are active.

Continued on next page

Test the ICF service */sap/public/icf_info/urlprefix*.

- a) Call the Maintain Services transaction SICF. Accept the default values and choose *Execute*.
- b) In the *Maintain Service* view, navigate to the */sap/public/icf_info* node.
- c) Check that all of the services below this node are active (displayed in black). Activate any services that are inactive.
- d) Click the */sap/public/icf_info/urlprefix* node with the secondary mouse button and choose *Test Service*. The URL prefixes are displayed in (SAP internal) format as they are called from the SAP Web Dispatcher.

The list should also contain your external alias */ADM102/myInfo* (from a previous exercise).

2. As part of the setup, determine the HTTP port of the ABAP message server for your SAP system.
 - a) The required port is defined in the AS ABAP using the profile parameter *ms/server_port_<xx>*. You can determine the current value in one of the following ways:
 - Call transaction SMMS and choose *Goto → Parameters → Display*.
 - Execute report or transaction *RSPFPAR* with the search option **ms/server_port_*** on the central instance.
 - Use transaction ST11 on the central instance to display trace file *dev_ms* and find the word **port** in the first rows.
3. On your server, create directory *E:\SAPWebDisp##*, where ## is your group number.



Note: Do not use spaces in the directory name.

- a) If you have not already done so, log onto the Terminal Services Client (also known as the RDP Client) at operating-system level using the user **<sid>adm**.
- b) Open an Explorer session (for example, by clicking the secondary mouse button and choosing *Start → Explore*) and navigate to the *E* drive.
- c) Click the right-hand screen area with the secondary mouse button and choose *New → Folder*. Enter **SAPWebDisp##** as the name of the folder.

Continued on next page

4. Copy the installation package of the SAP Web Dispatcher from the training share to the new directory.
 - a) Copy file `sapwebdisp_<version>.sar` from directory `M:\ADM102_62\SAP Web Disp` to the new directory `E:\SAPWebDisp##`.
5. Unpack the installation package of the SAP Web Dispatcher.
 - a) Open a command prompt in the `E:\SAPWebDisp##` directory (by clicking the folder in the left-hand area of the Explorer window with the secondary mouse button and choosing *Command Prompt Here*) and executing the command `sapcar -xvf sapwebdisp_<version>.sar`.
6. Which release of the SAP Web Dispatcher is used on the training system?
 - a) Open a command prompt in the `E:\SAPWebDisp##` directory (by clicking the folder in the left-hand area of the Explorer window with the secondary mouse button and choosing *Command Prompt Here*) and executing the command `sapwebdisp -v`.

The information that you require then appears at the start of the data that is displayed (“patch number”). At the end of the list, all problems that are solved with the current release status are listed (with associated SAP Notes).
7. Start the SAP Web Dispatcher with the bootstrap option and configure it for your SAP system. Use the following entries:

	DEV Group	QAS Group
<i>Hostname of Message Server</i>	Full host name of your server, for example twdf0042.wdf.sap.corp	
<i>HTTP Port of Message Server</i>	8100	8110
<i>Unique Instance Number for SAP Web Dispatcher</i>	05	15
<i>HTTP port number for SAP Web Dispatcher</i>	8005	8015
<i>Create configuration for ...</i>	s(mall) system	

Continued on next page



Caution: Remember to note the password that is generated for the *icmadm* user.

- a) In your *E:\SAPWebDisp##* directory, use the command prompt to execute command **sapwebdisp -bootstrap** and enter the data from the table above as required. Note the resulting data carefully.



Hint: Do not close the command prompt once you have finished.

8. Open the profile file of the SAP Web Dispatcher that was created, and note how your entries have been implemented. What is the URL used to call the Web admin interface?
 - a) Open the file *E:\SAPWebDisp##\sapwebdisp.pfl* with the notepad.
 - b) You can now determine the profile parameters for which your entries were required.
 - c) The URL of the Web admin interface has the following structure:
http://<server with domain>:<SAP Web Dispatcher port>/<admin prefix>. The **<SAP Web Dispatcher port>** is specified using profile parameter *icm/server_port_0* and the **<admin prefix>** is specified by the *PREFIX* value in parameter *icm/HTTP/admin_0*.

Result

The SAP Web Dispatcher is configured and has started for your SAP system.

Continued on next page

Task 2: Using the SAP Web Dispatcher

Send requests that are distributed using the SAP Web dispatcher.

1. Call the ICF service `/ADM102/myInfo` using your SAP Web Dispatcher.



Note: If you have not completed the relevant task in the “Internet Communication Framework” lesson, use the ICF service `/sap/bc/icf/info` instead.

- a) In your local Web browser (in the training room), enter the URL `http://<server with domain>:<SAP Web Dispatcher port>/ADM102/myInfo` (example for the QAS group and the server twdf0042: `http://twdf0042.wdf.sap.corp:8015/ADM102/myInfo`).
- b) If you choose *Refresh* a number of times in the Web browser, you can observe that both instances are used for processing the requests.



Hint: The distribution of the requests is based on the number of dialog work processes configured for each instance.

2. Create a *Dialog* logon group that includes only the ABAP dialog instance. Define this logon group for the external alias `/ADM102/myInfo`.

Continued on next page

Note what happens when you call the ICF service `/ADM102/myInfo` using your SAP Web Dispatcher.

- a) Call transaction SMLG and choose *Create*. In the *Logon Group* field, enter **Dialog** and use the input help (F4) for the *Instance* field to select your dialog instance. *Copy* and *Save* your new entry.
- b) Call transaction SICF and choose *External Aliases* to switch to the *Maintain External Aliases* view. Navigate to the external alias `/ADM102/myInfo` and double-click this node. Switch to change mode and use the input help (F4) to enter the logon group *Dialog* that you just created in the *Load Balancing* field on the *Service Data* tab page. Save your changes and go back a screen.


If you wish, you can now test the ICF service `/sap/public/icf_info/logon_groups`. The logon groups linked to ICF nodes are then displayed.

- c) In your local Web browser (in the training room), enter the URL **`http://<server with domain>:<SAP Web Dispatcher port>/ADM102/myInfo`** (example for the QAS group and the server twdf0042: **`http://twdf0042.wdf.sap.corp:8015/ADM102/myInfo`**).
- d) If you choose *Refresh* a number of times in the Web browser, you can see that the requests are always processed by one dialog instance.



Note: In the standard system, it can take up to two minutes for the SAP Web Dispatcher to recognize the configuration change (SAP Web Dispatcher parameter `wdisp/auto_refresh`).

Continued on next page

3. Call the Web dynpro Java application `/webdynpro/dispatcher/local/WhoAmI/Show` using your SAP Web Dispatcher.
 - a) In your local Web browser (in the training room), enter the URL `http://<server with domain>:<SAP Web Dispatcher port>/webdynpro/dispatcher/local/WhoAmI/Show` (case-sensitive). Example for the QAS group and server twdf0042:
`http://twdf0042.wdf.sap.corp:8015/webdynpro/dispatcher/local/WhoAmI/Show`.
-  **Note:** If you choose *Refresh* a number of times in the Web browser, you can see that the requests are always processed by the same instance (since the application, like any other Web dynpro application, is stateful). The *Clear Cookies* function allows you to reset the load-balancing cookies in your Web browser. The request generated when you choose *Refresh* may be distributed to another instance by the SAP Web Dispatcher.

Result

You have learned about the SAP Web Dispatcher's mode of operation.

Task 3: Restart the SAP Web Dispatcher Using Operating System Tools

Start and stop the SAP Web Dispatcher using operating system tools.

1. Stop your SAP Web Dispatcher.
 - a) Determine the process ID (PID) of your SAP Web dispatcher at operating system level of your server.

You can find this number, for example, from the data that appears when you start the SAP Web Dispatcher or from the trace file (default name `dev_webdisp`).
 - b) Open another command prompt (in Microsoft Windows, choose *Start* → *Run...* and enter `cmd`).
 - c) Run the command `sapntkill -INT <PID>` with the process ID that you have just identified.
 - d) Wait until the SAP Web Dispatcher displays the message “*** SAP Web Dispatcher shutdown completed (pid: <PID>) ***”.

Continued on next page

2. Start the SAP Web Dispatcher with the generated profile file.
 - a) In the `E:\SAPWebDisp##` directory, execute the command **sapwebdisp pf=sapwebdisp.pfl**.
 - b) Wait until the SAP Web Dispatcher displays the message “*** SAP Web Dispatcher up and operational (pid: <PID>) ***”.

Result

You can start and stop the SAP Web Dispatcher at operating system level.

Task 4: Optional: Set Up as Windows Service

Set up the SAP Web Dispatcher as a Windows service that is started automatically.

1. Stop the SAP Web dispatcher using operating system tools.
 - a) In a command prompt, execute the command **sapntkill -INT <PID>** with the current process ID.

Continued on next page

2. Configure your SAP Web Dispatcher as Windows service *SAPWebDisp##* and start it. This service should start automatically when you boot the host.

- a) In a command prompt, execute the following command: **ntscmgr install SAPWebDisp## -b "E:\SAPWebDisp##\sapweb-disp.exe" -p "service pf=E:\SAPWebDisp##\sapweb-disp.pfl"**, where ## is your group number.

The message "CreateService SUCCESS" appears.



Note: Make sure there are no spaces in the path.

- b) Open the Windows Service Manager (by choosing *Start* → *Programs* → *Administrative Tools* → *Services*). Select your *SAPWebDisp##* service with the secondary mouse button and choose *Start*.

You can monitor the start of the SAP Web Dispatcher using the trace file (*E:\SAPWebDisp##\dev_webdisp* in the standard system).

- c) In the Windows Service Manager, select your *SAPWebDisp##* service with the secondary mouse button and choose *Properties*.

On the *General* tab page under *Startup Type*, choose *Automatic* and then *OK*. The next time that you start the host (which you do not have to do immediately), your SAP Web Dispatcher will start automatically.

Result

You can configure the SAP Web Dispatcher under Windows as a service.

Task 5: Web Admin Interface

Use the Web-based interface for administration and monitoring of the SAP Web Dispatcher.


1. Call the Web admin interface using a Web browser.
 - a) Enter the URL that you previously noted in the Web browser.
With the standard values, this is **http://<server with domain>:<SAP Web Dispatcher port>/sap/wdisp/admin** (example for group QAS and server twdf0042:
http://twdf0042.wdf.sap.corp:8015/sap/wdisp/admin).
 - b) In the dialog box, enter the **icmadm** user and the generated password (see task 1).

Continued on next page

2. Change the password for the *icmadm* user.
 - a) Navigate to the menu option *HTTP Handler* → *Admin Handler*.
 - b) Click the *icmauth.txt* field with the secondary mouse button and choose *Change Users*.
 - c) Select the *icmadm* user and choose *Edit User*.
 - d) Now enter the previous password and the new password twice. Then choose *Save*.

You are then prompted to enter the new password on the Web admin interface.
3. Determine which servers are available for which logon groups.
 - a) Navigate to the menu option *Dispatching Module* → *Monitor Server Groups*. On the right of the screen, you can display details for each logon group (such as the next instance to be used).

Continued on next page

4. **Optional:** Use the *icmon* program to create another administration user **WebDispADM##** with a password of your choice.
- a) In a command prompt from directory *E:\SAPWebDisp##*, execute the command **icmon -a**.
-  **Note:** You do not have to enter the exact path to the executable *icmon* in the training environment (*PATH* is set to variable).
- If you operate the SAP Web Dispatcher on an independent host, you must copy the *icmon* program to this host.
- b) Confirm the default file name *icmauth.txt* by pressing *ENTER*.
- c) In the menu, choose *Add User to Set* with the command **a**. Make the following entries:

<i>User name:</i>	WebDispADM##
<i>(Re-)Enter Password</i>	Any (at least four characters)
<i>Group name</i>	admin
<i>Subject value of client cert</i>	Leave empty

- d) You can then display the list of administrators for the SAP Web Dispatcher by choosing *List Users of Set*, command **l**.
- e) Save your changes with the *icmon* command **s** and end the program with **q**.
- f) Call the Web admin interface in your Web browser again and logon with the new user.

Result

You can now configure and use the easy Web admin interface of the SAP Web Dispatcher.



Lesson Summary

You should now be able to:

- Outline the function of the SAP Web Dispatcher
- Explain how you can use the SAP Web Dispatcher to distribute workload across the different instances of an SAP system

Related Information

- Online documentation for SAP NetWeaver 7.0: *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Solution Life Cycle Management by Key Capability* → *System Management* → *SAP Web Dispatcher*.
- SAP Note 538405: *Composite SAP Note: SAP Web Dispatcher*
- SAP Note 552286: *Troubleshooting for the SAP Web Dispatcher*
- SAP Note 499327: *Using the SAP Web Dispatcher with SAP Web AS 6.10*
- SAP Note 740234: *Cascade of SAP Web dispatchers*
- SAP Note 864878: *Access restrictions in ICM and SAP Web Dispatcher 7.00*
- SAP Note 1040325: *HTTP load balancing: Message Server or Web Dispatcher?*

Lesson: Load Balancing in the SAP NetWeaver AS Java Environment

Lesson Overview

An SAP system can be scaled using the number of application servers and the number of dialog work processes (ABAP) or server processes (Java) for each instance. Requests to the SAP system should be distributed as equally as possible across all application servers and processes. A load balancing procedure is required to do this. The techniques used for this in the SAP system are introduced in this lesson.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain how load balancing can be realized in the SAP system

Business Example

With large applications, it is best to spread the load across several components. Load balancing is also possible with SAP NetWeaver AS Java.

Overview

In this section, the different mechanisms for load balancing that are available for SAP NetWeaver AS are presented. You can essentially differentiate between two mechanisms for load balancing: client-based and server-based load balancing. In general, we recommend server-based load balancing.

Server-Based Load Balancing

A load balancer connected in front acts as a central entry point to the SAP system. This is the case, even if the SAP system is made up of multiple application servers. These technique offers the following advantages:

- All application servers can be addressed using a common IP address or a common name.
- The users always use the same URL to access the system.
- One SSL server certificate is sufficient for all of the application servers.
- The advantages listed above reduce the operating and maintenance effort and costs.

This central entry point to the SAP system can be realized using an additional component, known as a “load balancer”. As shown in the following figure, this load balancer receives inbound requests and distributes these to the application servers.

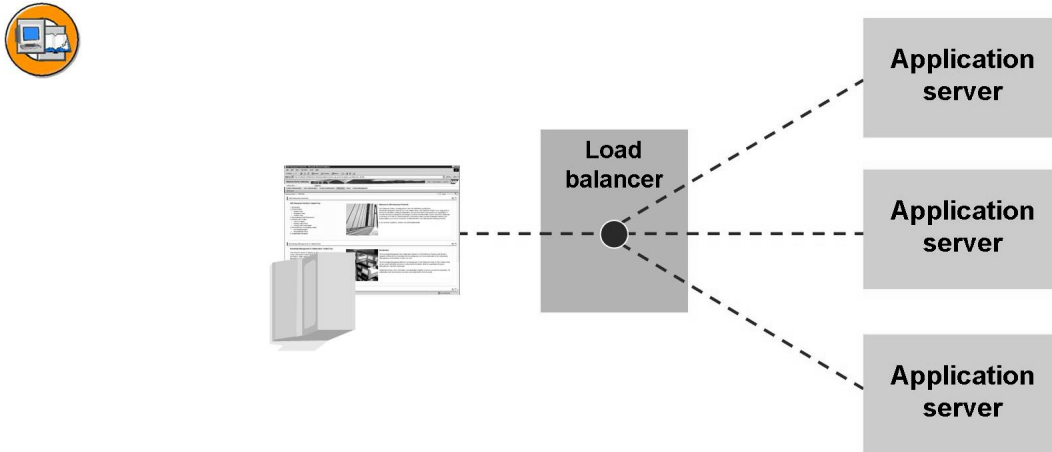


Figure 30: Server-Based Load Balancing

client-based load balancing

In addition to SAP's preferred method of server-based load balancing, there are other methods, which can be preferred in certain circumstances. In particular, if a simple implementation of load balancing is desired. With this client-based load balancing, all inbound client requests are initially directed to a central location in the system, a load balancing server, when the connection is first made. The load balancing server informs the client which application server it should address. This is shown in the following figure.

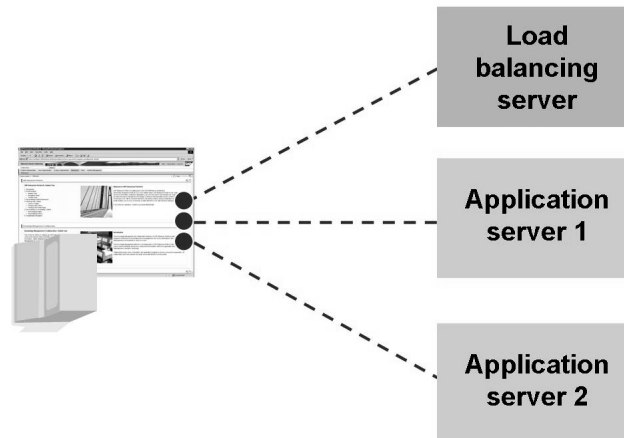


Figure 31: Client-Based Load Balancing

Client-based load balancing can be realized using the following mechanisms:

Rerouting the requests using functions -

- That provide the HTTP protocol (redirect)
- That provide the Domain Name System (DNS) protocol, with which the namespace in the Internet is managed

A simple method of load balancing is already implemented in SAP NetWeaver AS, based on the rerouting of HTTP requests. Their function is shown in the following figure.

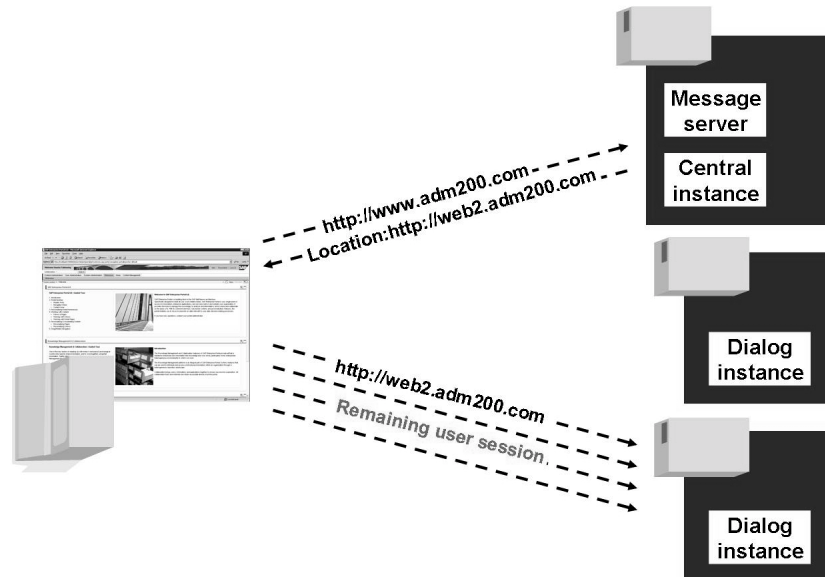


Figure 32: Load Balancing Using the SAP Message Server

This mechanism functions as follows:

1. The browser sends a request to the message server.
2. The message server returns the address of an appropriate application server to the browser (redirect).
3. The browser now sends a request to this application server.
4. The user remains connected to this application server for the rest of the duration of the session.

Although this method is implemented in SAP NetWeaver AS using the message server of the central instance, and is already available after installation, it is not the preferred method due to a number of disadvantages. Some of these disadvantages are listed briefly here:

- Can lead to confusion of the user, since the URL displayed in the browser changes with the rerouting
- If Favorites are created in the browser, these point to the server to which the user was rerouted
- Each application server requires a server certificate
- Can cause problems if a firewall is used

Stateless and Stateful Web Applications

The programming model that underlies the development of Web applications has an important influence on a load balancer. The programming model differentiates between “stateless” and “stateful” Web applications.

The programming model for stateless requests is used for simple applications, for which each request to SAP NetWeaver AS is independent of all other requests.

The programming model for stateful requests is used for more complex applications, which are based on a transactional concept. With these applications, information about the status of the user session must be stored in the application server.

The mechanism for load balancing in the SAP system must support both stateless and stateful requests. Stateful requests are a particular challenge for the load balancer, since the HTTP protocol only supports stateless requests. This is illustrated in the following figure. The first request is forwarded to an application server by the load balancer. If a subsequent request is forwarded to a different application server, this has no information about the user context.

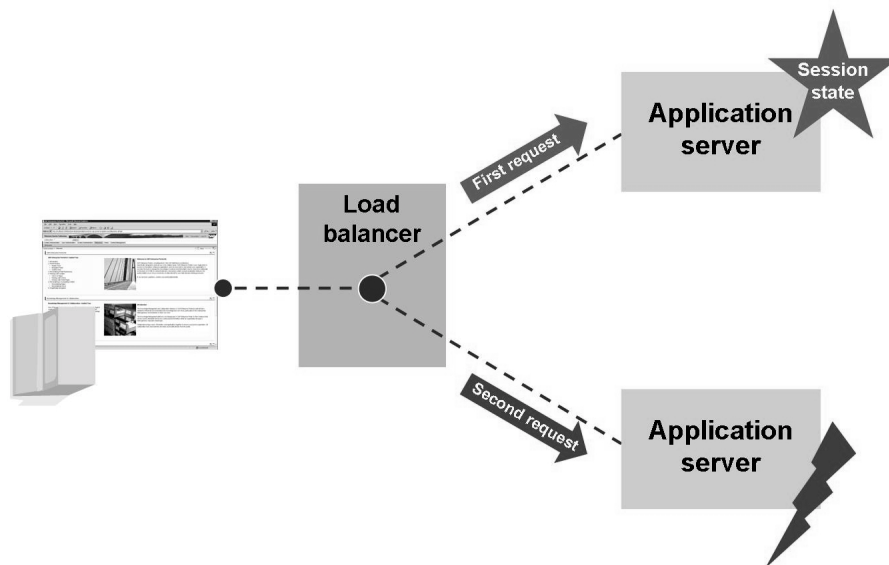


Figure 33: Stateful Requests

The load balancer must therefore ensure that stateful requests are always forwarded to the same application server. This can be achieved by different implementations in the load balancer. However, these different techniques are not presented in more detail here.

Realization of Load Balancing in SAP NetWeaver AS Java

After these initial considerations about load balancing, the realization in SAP NetWeaver AS Java is now presented in this section.

Load balancing within SAP NetWeaver AS Java allows the optimal distribution of the incoming requests to the available resources. SAP NetWeaver Application Server provides load balancing at different levels, as shown in the following figure.

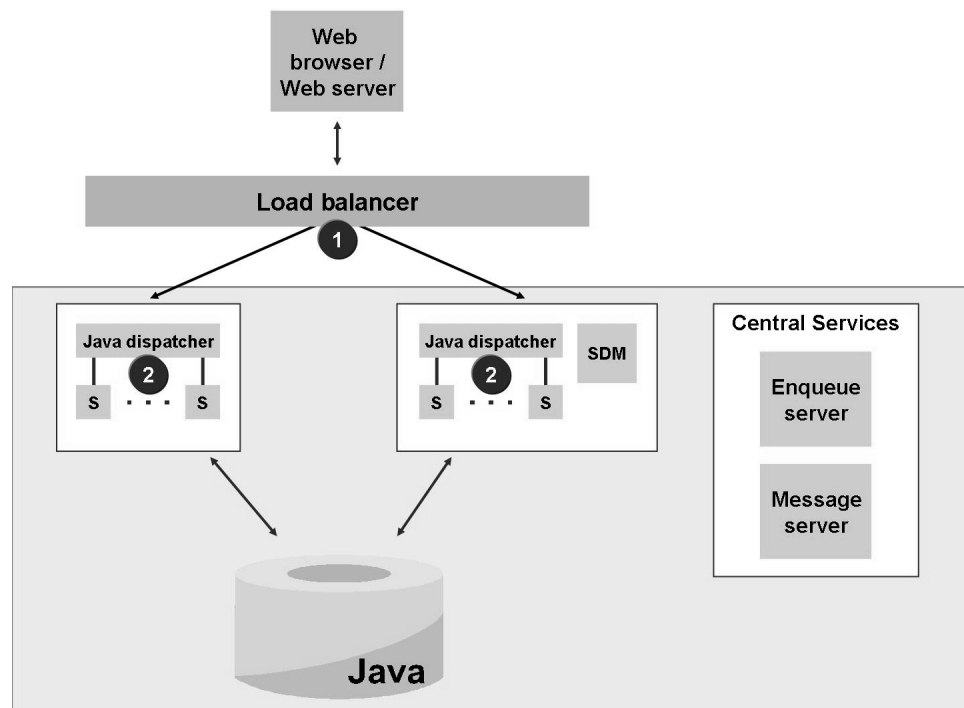


Figure 34: Load balancing in SAP NetWeaver AS Java

In a cluster with multiple SAP NetWeaver AS Java instances, load balancing is performed using a load balancer connected in front (1). Within the Java instance, the Java dispatcher (2) distributes the inbound requests to the server processes with which it is connected.

Load Balancing Between Many Java Instances

The following figure shows a system with multiple Java dispatchers, with a SAP Web Dispatcher connected in front of them in the DMZ as a load balancer. This performs the load balancing between the Java dispatchers, which then distribute the requests to their server processes.



Hint: You can also use any other load balancing device instead of the SAP Web Dispatcher. In this case, you need to register the servers and ports with it; the communication with the message server and the mapping to SAP logon groups does not take place.

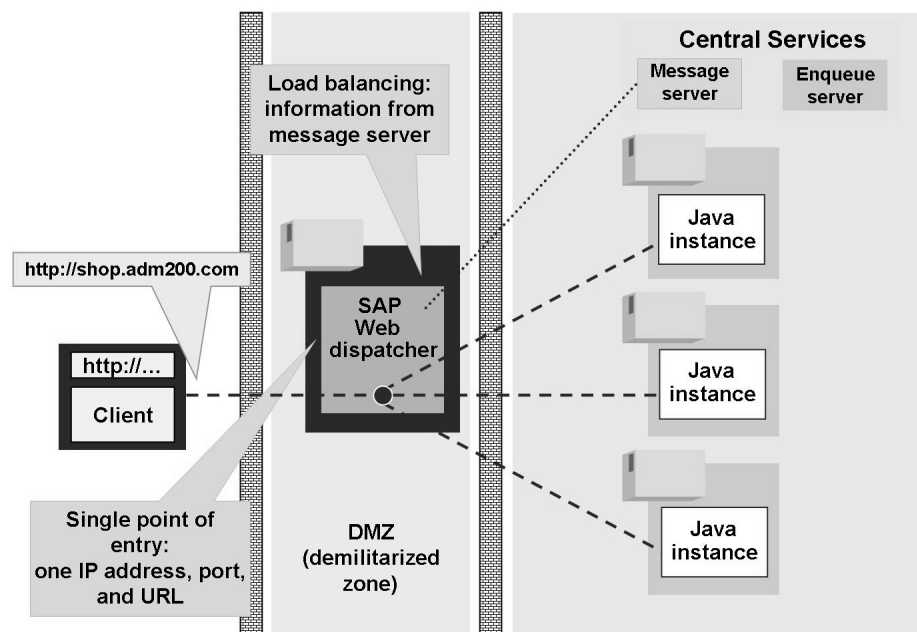


Figure 35: Load Balancing Between Many Java Instances

The SAP Web Dispatcher fetches the information that it requires from the message server about:

- All Java dispatchers with their HTTP ports, to which it can forward requests
- The capacities of the connected Java instances, so that it can use the weighted round robin procedure. For this, the SAP Web Dispatcher simply needs, in its profile file, the port at which it can reach the message server (parameter `ms/http_port`).

The SAP Web Dispatcher is delivered with Central Services (enqueue service and message service). In the standard installation, you will find this in the directory `/usr/sap/<SID>/SCS01/exe`.

The SAP Web Dispatcher can be used for load balancing in the following scenarios:

- Java-only scenario, as described here.
- ABAP-only scenario (see SAP customer training course ADM102, "SAP NetWeaver AS Administration II")
- ABAP-only scenario (see SAP customer training course ADM102, "SAP NetWeaver AS Administration II")

Appendix: SAP Web Dispatcher

As previously described, the SAP Web Dispatcher, which lies between the Internet and the SAP system, can be used as a load balancer. It is the entry point for HTTP(S) requests into your system, which consists of one or more Web application servers. As a "software Web switch", it can reject or accept connections. When it accepts a connection, it distributes the requests to ensure an even distribution across the servers (load balancing).



Hint: Not only does using the SAP Web Dispatcher allow you to realize load balancing across multiple SAP NetWeaver AS instances, it also provides security functions (entry point in the DMZ, SSL, URL filtering).

The SAP Web Dispatcher forwards inbound requests (HTTP, HTTPS) to the SAP NetWeaver AS instances of the SAP system in turn, where the number of requests that a SAP Web AS receives is weighted according to its capacity. The capacity of a SAP NetWeaver AS ABAP depends on the number of configured dialog work processes. For SAP NetWeaver AS Java, the capacity is determined by the number of server processes. If the application is stateful, the SAP Web Dispatcher ensures at the next request that the user is again forwarded to the server processing his or her application. It uses the session cookie to do this for HTTP connections, and the client IP address for end-to-end SSL. The SAP Web Dispatcher also decides whether the inbound request is to be forwarded to a SAP NetWeaver AS ABAP or a SAP NetWeaver AS Java.



Hint: Unlike the HTTP load balancing performed by the SAP message server, no redirect is performed when using the SAP Web Dispatcher. In this way, the associated disadvantages (a large number of IP addresses must be known, bookmarking is not possible, authentication after a change of application server) are also avoided.

The SAP Web Dispatcher is a separate program that can run on a host that is directly connected to the Internet. It requires minimal configuration. You only need to enter the following data in the profile file for the SAP Web Dispatcher:

- Port on which the HTTP(S) requests are to be received (parameter `icm/server_port_<xx>`)
- Host and HTTP port of the SAP Message server (parameter `rdisp/mshost` and parameter `ms/http_port`)

If you want to be able to call the Web application externally, for example with the URL `www.adm200.com`, this host name must be mapped internally to the SAP Web Dispatcher. This then forwards the HTTP(S) request to a suitable SAP NetWeaver AS.



Hint: The SAP Web Dispatcher is presented in detail in the SAP customer training course ADM102, “SAP NetWeaver AS Administration II”. For information about the change history of the SAP Web Dispatcher, see the Composite SAP Note on the SAP Web Dispatcher (SAP Note 538405).



Lesson Summary

You should now be able to:

- Explain how load balancing can be realized in the SAP system



Unit Summary

You should now be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the areas of use of SAP ITS, ICM, AS ABAP, and AS Java
- Describe the architecture of the SAP ITS (standalone)
- Perform simple administrative tasks on an SAP ITS
- Describe the implementation area of the ICM
- Configure and monitor the ICM
- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Activate and use the integrated ITS as of AS ABAP 6.40
- Outline the function of the SAP Web Dispatcher
- Explain how you can use the SAP Web Dispatcher to distribute workload across the different instances of an SAP system
- Explain how load balancing can be realized in the SAP system



Test Your Knowledge

1. Which of the following technology components can be used together with an SAP ERP Central Component (ECC 6.0) system?

Choose the correct answer(s).

- ☐ A SAP Internet Transaction Server (SAP ITS), standalone
- ☐ B SAP Internet Transaction Server (SAP ITS), integrated
- ☐ C Internet Communication Manager (ICM)
- ☐ D Web Dynpro ABAP
- ☐ E Web Dynpro for Java

2. Which tools can you use to administer an SAP ITS standalone?

Choose the correct answer(s).

- ☐ A WGate configuration tool
- ☐ B Transaction SITS in the SAP system
- ☐ C ITS Administration Tool
- ☐ D Microsoft Management Console with the SAP ITS Snap-In

3. Which statement(s) is/are correct?

Choose the correct answer(s).

- ☐ A The ICM is implemented as a thread and is available for a large number of operating systems.
- ☐ B You can use an instance profile parameter to configure how many ICMs are started for each ABAP dispatcher.
- ☐ C SAP recommends that you operate a separate ICM for each client in an SAP system.

4. The ICM is relevant only for applications based on BSPs.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

5. Which statement(s) is/are correct?

Choose the correct answer(s).

- ☐ A The SAP Web Dispatcher stores system information in a small local database.
- ☐ B The SAP Web Dispatcher knows the capacity of all application servers in the SAP system.
- ☐ C The SAP Web Dispatcher makes a firewall superfluous.
- ☐ D The SAP Web Dispatcher communicates with the message server.

6. Which software components allow load balancing in the context of SAP NetWeaver AS Java?

Choose the correct answer(s).

- ☐ A SAP Web Dispatcher
- ☐ B ABAP dispatcher
- ☐ C Java dispatcher
- ☐ D Server processes



Answers

1. Which of the following technology components can be used together with an SAP ERP Central Component (ECC 6.0) system?

Answer: B, C, D, E

The technical basis of SAP ECC 6.0 is SAP NetWeaver Application Server 7.00 (part of SAP NetWeaver 7.0). The standalone version of SAP ITS is released only in conjunction with systems based on SAP Web AS 6.40 or earlier. All other technologies can, in principle, be used:

- The integrated SAP ITS is available as of AS ABAP 6.40.
- ICM is available as of AS ABAP 6.10.
- Web Dynpro for ABAP is available as of AS ABAP 7.00.
- Web Dynpro for Java is available as of AS ABAP 6.40.

2. Which tools can you use to administer an SAP ITS standalone?

Answer: A, C

You manage an SAP ITS standalone with browser-based tools.

- WGate configuration tool (for WGates)
- ITS administration tool (for AGates)

3. Which statement(s) is/are correct?

Answer:

All of the answers provided are incorrect. The ICM is a process (which internally consists of multiple threads) that is available for all operating systems supported by SAP. You can use a profile parameter to configure either no ICM or one ICM for each SAP instance (and therefore for each ABAP dispatcher). Scaling is performed using the settings for the number of threads and memory allocation.

4. The ICM is relevant only for applications based on BSPs.

Answer: False

Executing BSP applications requires the ICM. However, there are also other applications (Web services such as SOAP/XML or Web reporting with the SAP BW), that use the ICM and ICF but do not use BSPs). Thanks to the ICM and ICF, the AS ABAP can also perform the role of Web client and can process other protocols such as SMTP in addition to HTML.

5. Which statement(s) is/are correct?

Answer: B, D

The SAP Web Dispatcher communicates with the message server by HTTP(S) (depending on the installation variant with the ABAP or Java message server) to obtain information about the available application servers (including their capacity). Although the SAP Web Dispatcher can act as a URL filter, it does not replace the function of a firewall. Only requests that are addressed to the configured port of the SAP Web Dispatcher are processed.

6. Which software components allow load balancing in the context of SAP NetWeaver AS Java?

Answer: A, C

The SAP Web Dispatcher distributes inbound requests across multiple SAP NetWeaver AS Java instances. The Java dispatcher receives the requests within an instance and distributes these to the server processes of the instance.

Unit 2

Basics of User Administration AS ABAP

Unit Overview

This unit provides information about the basics of user administration in SAP NetWeaver AS ABAP. Therefore, all of the terms used in this unit refer to SAP NetWeaver AS ABAP. In the next unit, you will learn about the user and authorization concept of SAP NetWeaver AS Java or SAP NetWeaver AS ABAP and Java. Terms such as “role” or “group” will have different meanings there. Therefore, to make a clear distinction between the terminology used in both units, the role introduced here is described as a “PFCG role” or “ABAP role” and the groups introduced here are described as “ABAP groups”.

This unit covers the authorization concept of SAP NetWeaver AS ABAP. The focus is on explaining important terms and creating roles and authorization profiles. System parameters will be used to demonstrate how relevant settings can be made in the system. The use of central directory services is also introduced.



Unit Objectives

After completing this unit, you will be able to:

- Create users
- Copy, create, and maintain roles
- Maintain the assignment of roles and users
- Set system parameters for user logons
- Name standard users in the SAP system
- Locate authorization problems
- Describe the concept of Central User Administration
- Describe connection to directory services

Unit Contents

Lesson: User Administration Concept.....	115
Exercise 4: Fundamentals of User Administration	121

Lesson: Authorization Concept	126
Exercise 5: Working with Roles.....	135
Lesson: Login Parameters and User Info.....	141
Lesson: Appendix: Advanced User Administration Topics	151

Lesson: User Administration Concept

Lesson Overview

This lesson explores the administration of user master records. Creating, copying, and maintaining master records of this type will be described in more detail.



Lesson Objectives

After completing this lesson, you will be able to:

- Create users

Business Example

The users of the SAP system require their own user with appropriate authorizations to log on. The administrator sets up a user ID in the system for each user.

Basics of User Administration

The concept of the user master record and the authorization concept are explained in more detail below. Both of these are important to obtain a better understanding of SAP systems.

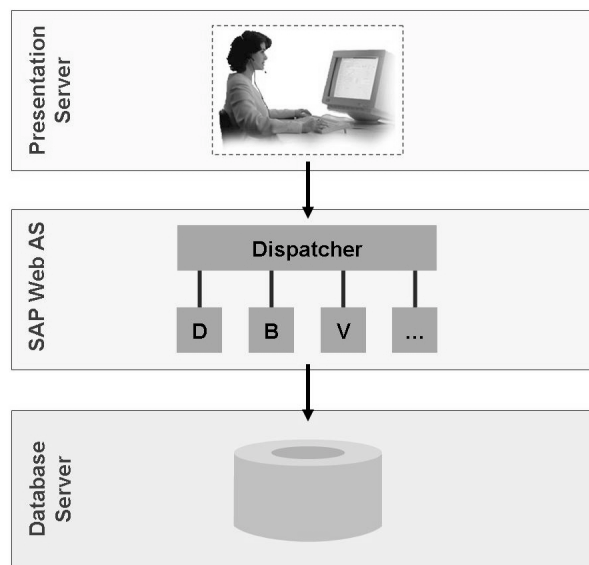


Figure 36: Users in the SAP Environment

The term **user** usually means user ID here. People log on to an operating system, a database, or an SAP system using a user/password combination. Operating systems, databases, and SAP systems usually have different authorization concepts. If a user/password combination is created in an SAP system for a person, this does not mean that it is possible to log on to the operating system of a host with the same user/password combination. However, it is possible that identical user/password combinations are created for SAP systems and operating systems.

➔ **Note:** User requests are processed by SAP work processes. These work processes all use a common user to access the database.

This unit deals exclusively with SAP users, which people use to log on to a client of an SAP system. Users and authorization data are client-dependent.

Access to the operating system level of the SAP Web Application Server and the database server must be protected, or the operation or the data of the SAP systems could be threatened.

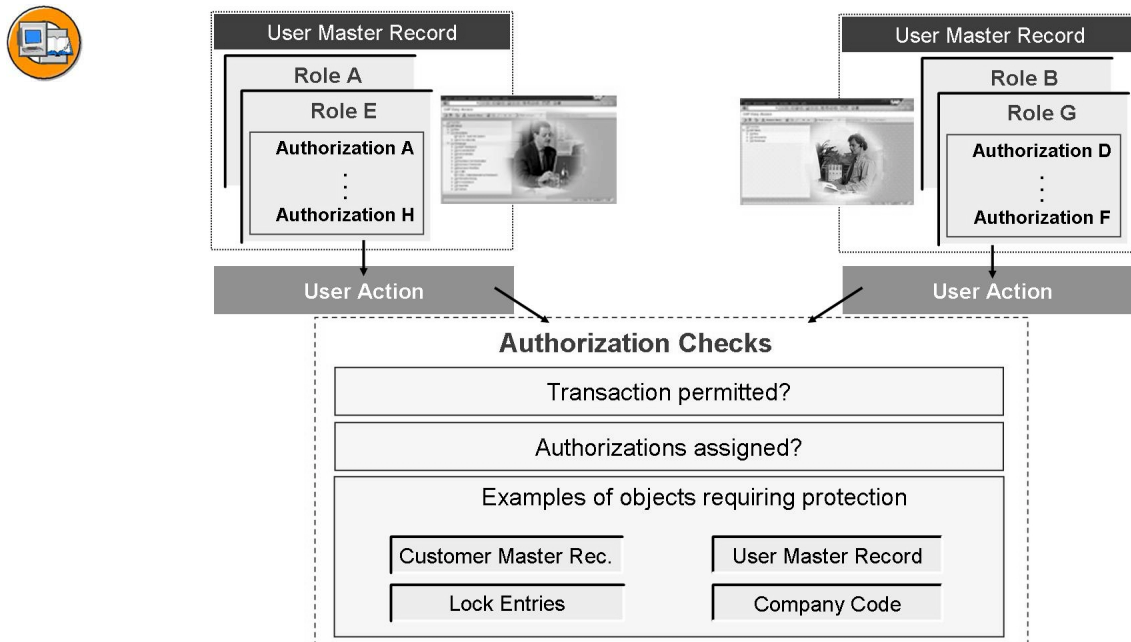


Figure 37: Users and Authorizations

A person can log on to a client of an SAP system if they know the user/password combination for a user master record.

In the SAP system, there is an authorization check every time a transaction is called. If a user attempts to start a transaction for which he or she is not authorized, the system rejects the user with an appropriate error message.

If the user starts a transaction for which he or she has authorization, the system displays the initial screen of this transaction. Depending on the transaction called, the user enters data and performs actions on this screen. Additional authorization checks are made for data and actions that are to be protected.

Users are assigned authorizations using roles. The authorizations are combined in roles and the roles are entered in the user master record. This is explained in more detail later in this unit.

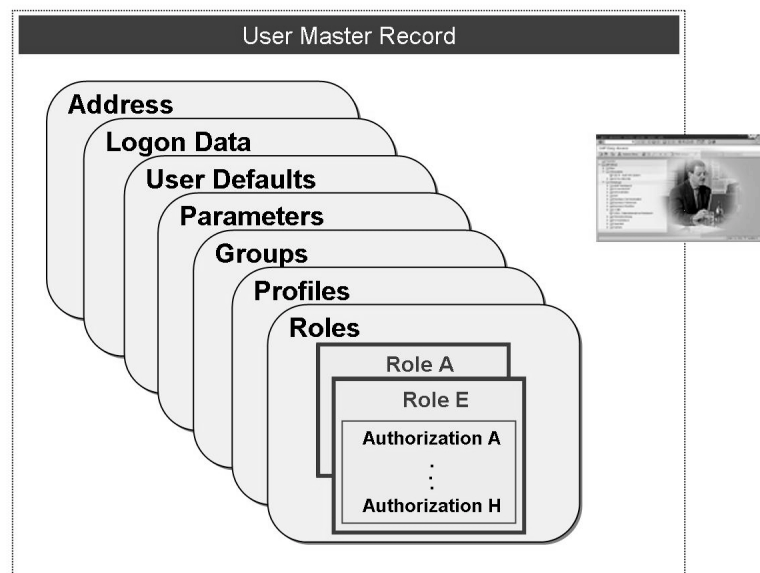


Figure 38: User Master Record

The user type is an important property of a user. Different user types are available for different purposes:

Dialog

A normal *dialog* user is used for all logon types by just one person. During a dialog logon, the system checks for expired/initial passwords, and the user has the opportunity to change his or her own password. Multiple dialog logons are checked and, if appropriate, logged.

System

Use the *System* user type for dialog-free communication within a system or for background processing within a system, or also for RFC users for various applications, such as ALE, Workflow, Transport Management System, Central User Administration. It is not possible to use this type of user for a dialog logon. Users of this type are excepted from the usual settings for the validity period of a password. Only user administrators can change the password.



Note: See also SAP Note 622464.

Communication

Use the *communication* user type for dialog-free communication between systems. It is not possible to use this type of user for a dialog logon. The usual settings for the validity period of a password apply to users of this type.

Service

A user of the type *Service* is a dialog user that is available to a larger, anonymous group of users. In general, you should only assign highly restricted authorizations to users of this type. Service users are used, for example, for anonymous system accesses using an ITS or ICF service. The system does not check for expired/initial passwords during logon. Only the user administrator can change the password. Multiple logons are permitted.

Reference

Like the service user, a *reference* user is a general non-person-related user. You cannot use a reference user to log on. A reference user is used only to assign additional authorizations. You can specify a reference user for a dialog user for additional authorization on the *Roles* tab page.



User Type	SAP GUI-compatibility	
	SAP GUI-compatible	Not SAP GUI-compatible
Dialog	Dialog	Communication
Communication		
System		
Service	Service	System
Reference		

Figure 39: User Types

To start user maintenance (transaction SU01), choose *Tools → Administration → User Maintenance → Users*.

You can create a new user master record by copying an existing user master record or creating a completely new one. The user master record contains all data and settings that are required to log on to a client of the SAP system. This data is divided into the following tab pages:

- *Address*: Address data
- *Logon data*: Password and validity period of the user, and user type. For further information about the password rules for special users, refer to SAP Note 622464,
- *Defaults*: Default values for a default printer, the logon language, and so on,
- *Parameters*: User-specific values for standard fields in SAP systems,
- *Roles and Profiles*: Roles and profiles that are assigned to the user,
- *Groups*: For the grouping of users for mass maintenance.

You must maintain at least the following input fields when creating a user: **Last name** on the *Address* tab page, **initial password** and identical **repetition of password** on the *Logon Data* tab page.

Exercise 4: Fundamentals of User Administration

Exercise Objectives

After completing this exercise, you will be able to:

- Create users

Business Example

As an administrator, you need to create new users and lock existing users.

Task 1: Creating Users

Create a user in client 100 with the name ADMIN<##>, where <##> is your group number. Follow the steps described below.

1. Log on to client 100 in your SAP system and create a user (master record) with the name ADMIN<##>.
2. Maintain the first and last names of the user.
3. Assign the user an initial password (PW: _____). Make sure you use the correct upper and lower case. Assign it to the *User Group for Authorization Check* **SUPER**.
4. Enter a default value for the logon language for the user (such as EN or DE).
5. Save the user master record.

Task 2: Creating a Special User



Caution: You should perform the following task in client “000”.

To avoid confusion, log off all users that are currently logged on to client **100** completely.

After you have logged on to client 000, create the user CSMREG in a particular way.

This user is needed for cross system monitoring.

1. In transaction RZ21, use the path *Technical Infrastructure → Configure Central System → Create CSMREG User*. Assign the password **monitor**.

In transaction SU01, make sure that the user CSMREG was created correctly.

Continued on next page

This user requires the role *SAP_BC_CSMREG* and the corresponding profile.

2. The user you have just created is required for a later exercise. After you have successfully created the user, log off completely from client 000 again.

Solution 4: Fundamentals of User Administration

Task 1: Creating Users

Create a user in client 100 with the name ADMIN<##>, where <##> is your group number. Follow the steps described below.

1. Log on to client 100 in your SAP system and create a user (master record) with the name ADMIN<##>.
 - a) Start transaction SU01. Enter the name **ADMIN<##>** in the *User* field and choose *Create*.
2. Maintain the first and last names of the user.
 - a) Choose the *Address* tab page. Enter the names in the appropriate fields.
3. Assign the user an initial password (PW:_____). Make sure you use the correct upper and lower case. Assign it to the *User Group for Authorization Check* **SUPER**.
 - a) Choose the *Logon Data* tab page. In the *Initial Password* field, enter the password and press the tab key to move to the *Repeat Password* field. Enter the password again in this field. Move the cursor to the *User Group* input field, and choose the group **SUPER** from the F4 help by double-clicking on it.
4. Enter a default value for the logon language for the user (such as EN or DE).
 - a) Choose the *Defaults* tab page. In the *Logon Language* field, enter **EN** for English, or **DE** for German, for example.
5. Save the user master record.
 - a) Choose *Save*.

Continued on next page

Task 2: Creating a Special User



Caution: You should perform the following task in client “000”.

To avoid confusion, log off all users that are currently logged on to client **100** completely.

After you have logged on to client 000, create the user CSMREG in a particular way.

This user is needed for cross system monitoring.

1. In transaction RZ21, use the path *Technical Infrastructure → Configure Central System → Create CSMREG User*. Assign the password **monitor**.

In transaction SU01, make sure that the user CSMREG was created correctly.

This user requires the role *SAP_BC_CSMREG* and the corresponding profile.

- a) Follow the exercise text.
2. The user you have just created is required for a later exercise. After you have successfully created the user, log off completely from client 000 again.
 - a) You can log off from a system quickly and easily by entering **/nex** in the command field of the standard toolbar (second bar from the top) and confirming this with *Enter*. This closes all sessions at once and logs you off from the system.

Result

You have now created the user ADMIN<##> (in client 100) and CSMREG (in client 000).



Lesson Summary

You should now be able to:

- Create users

Lesson: Authorization Concept

Lesson Overview

In this lesson, the terms authorization object, authorization profile, authorization check, and role are discussed in a common context. The focus here is on role maintenance; that is, on creating a role.



Lesson Objectives

After completing this lesson, you will be able to:

- Copy, create, and maintain roles
- Maintain the assignment of roles and users

Business Example

The authorizations for users are created using roles and profiles. Administrators create the roles, and the system supports them in creating the associated authorizations.

Authorization Objects and Authorization Checks

Understanding the SAP authorization concept requires knowledge of the role and the authorization profile in the user master record. This lesson provides you with the necessary knowledge to be able to create your own roles and authorizations.

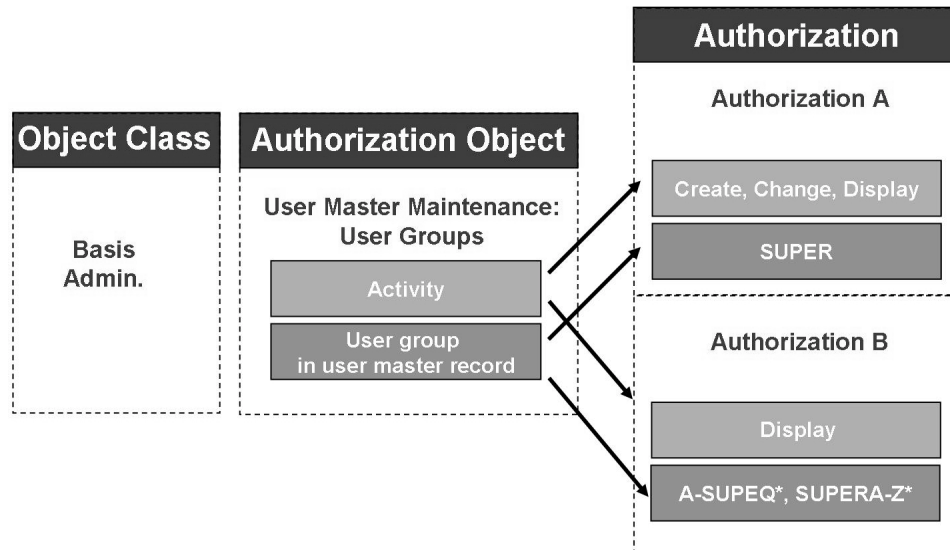


Figure 40: Authorization Objects

Actions and the access to data are protected by authorization objects in the SAP system. The authorization objects are delivered by SAP and are in SAP systems. To provide a better overview, authorization objects are divided into various object classes.

Authorization objects allow complex checks that involve multiple conditions that allow a user to perform an action. The conditions are specified in authorization fields for the authorization objects and are AND linked for the check. Authorization objects and their fields have descriptive and technical names. In the example in the figure, the authorization object "User master maintenance: User Groups" (technical name: *S_USER_GRP*) contains the two fields "Activity" (technical name: *ACTVT*) and "User Group in User Master" (technical name: *CLASS*). The authorization object *S_USER_GRP* protects the user master record. An authorization object can include up to ten authorization fields.

An authorization is always associated with exactly one authorization object and contains the value for the fields for the authorization object. An authorization is a permission to perform a certain action in the SAP system. The action is defined on the basis of the values for the individual fields of an authorization object. Example: Authorization B in the graphic for the authorization object *S_USER_GRP* allows the display of all user master records that are **not** assigned to the user group *SUPER*. Authorization A, however, allows records for this user group to be displayed.

There can be multiple authorizations for one authorization object. Some authorizations are delivered by SAP, but the majority are created specifically for the customer's requirements.

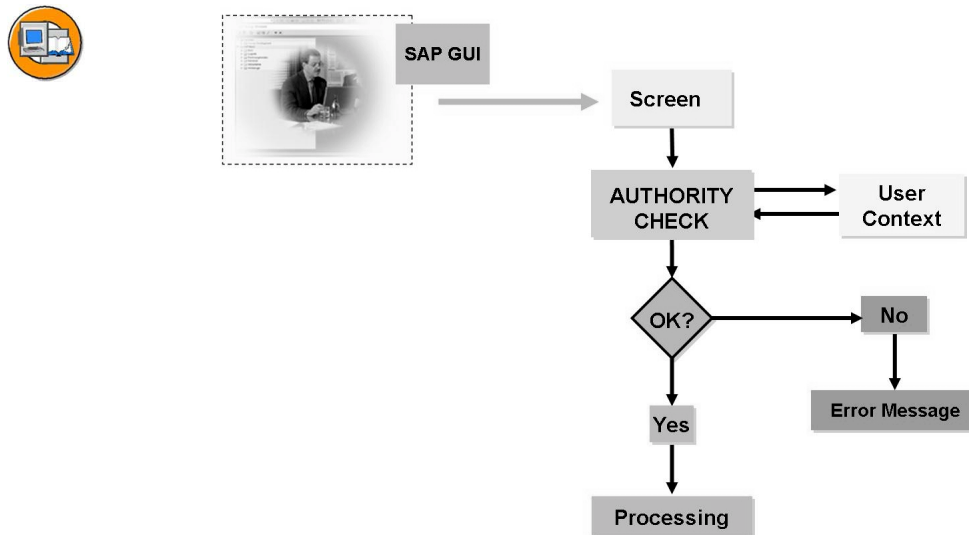


Figure 41: Authorization Check

When a user logs on to a client of an SAP system, his or her authorizations are loaded in the user context. The user context is in the user buffer (in the main memory, query using transaction code SU56) of the application server.

When the user calls a transaction, the system checks whether the user has an authorization in the user context that allows him or her to call the selected transaction. Authorization checks use the authorizations in the user context. If you assign new authorizations to the user, it may be necessary for this user to log on to the SAP system again to be able to use these new authorizations (for more information, see SAP Note 452904 and the documentation for the parameter *auth/new_buffering*).

If the authorization check for calling a transaction was successful, the system displays the initial screen of the transaction. Depending on the transaction, the user can create data or select actions. When the user completes his or her dialog step, the data is sent to the dispatcher, which passes it to a dialog work process for processing. Authority checks (*AUTHORITY-CHECK*) that are checked during runtime in the work process are built into the coding by the ABAP developers for data and actions that are to be protected. If the user context contains all required authorizations for the checks (return code = 0), the data and actions are processed and the user receives the next screen. If one authorization is missing, the data and actions are not processed and the user receives a message that his or her authorizations are insufficient. This is controlled by the evaluation of the return code. In this case, it is not equal to 0.

All authorizations are permissions. There are no authorizations for prohibiting. **Everything that is not explicitly allowed is forbidden.** You could describe this as a “positive authorization concept”.

Role Maintenance: Menus and Authorizations



Role Maintenance

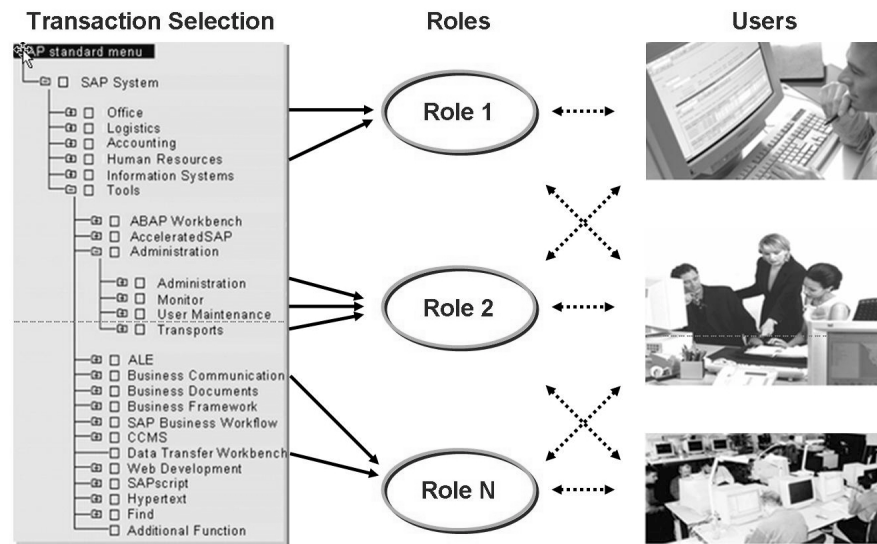


Figure 42: Role Maintenance

Role Maintenance (transaction PFCG, previously also called Profile Generator or activity groups) simplifies the creation of authorizations and their assignment to users. In role maintenance, transactions that belong together from the company's point of view are selected. Role maintenance creates authorizations with the required field values for the authorization objects that are checked in the selected transactions.

A role can be assigned to various users. Changes to a role therefore have an effect on multiple users. Users can be assigned various roles.

The user menu comprises the role menu(s) and contains the entries (transactions, URLs, reports, and so on) that are assigned to the user through the roles.

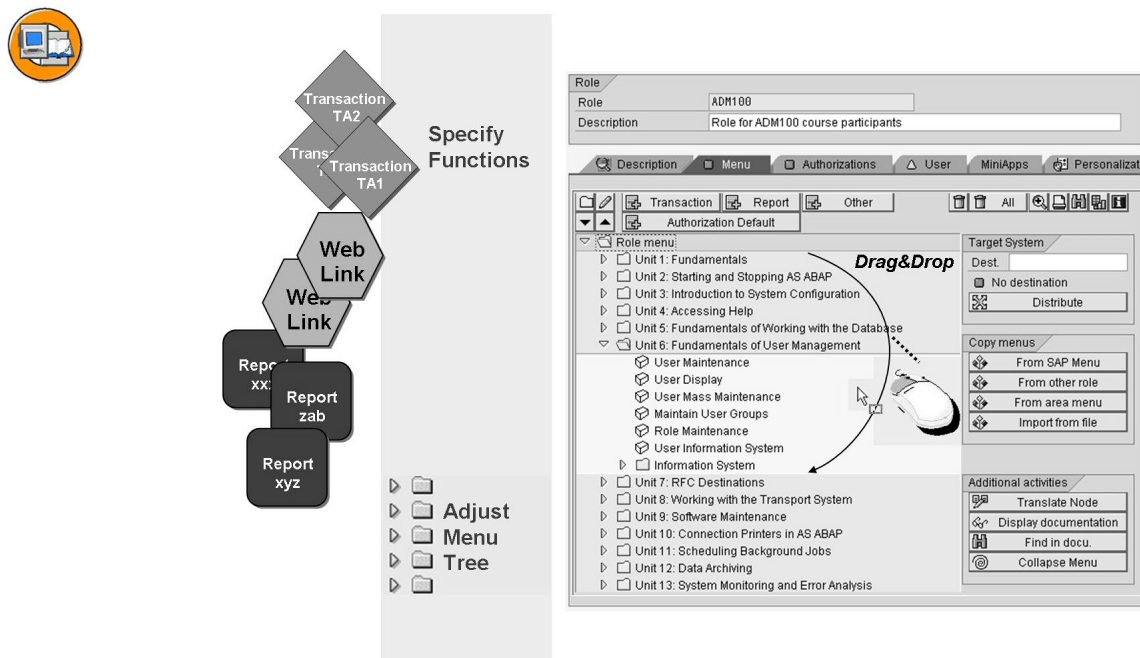


Figure 43: Menu Layout

You can access role maintenance with transaction PFCG or by choosing *Tools* → *Administration* → *User Maintenance* → *Role Administration* → *Roles*. Enter the name of the role and choose the icon for *Create* or *Change*. Choose the *Menu* tab page.

Select and change functions: The menu tree can be adjusted for the individual roles as required.

You can insert **transactions** into the tree structure or delete them from it.

By choosing the *Report* button, you can integrate **Reports**. In this case role maintenance creates transaction codes (if they do not already exist) with which the reports can be called.

By choosing the *Other* button, you can add **Internet addresses** or **links to files** (such as tables or text files). When integrating files, you must use the storage paths instead of URLs. You can also specify BW Web reports, and links to external mail systems and Knowledge Warehouse.

Change menus: You can create, move, delete, and rename directories and subdirectories as required. You can use the Drag&Drop function in role maintenance.

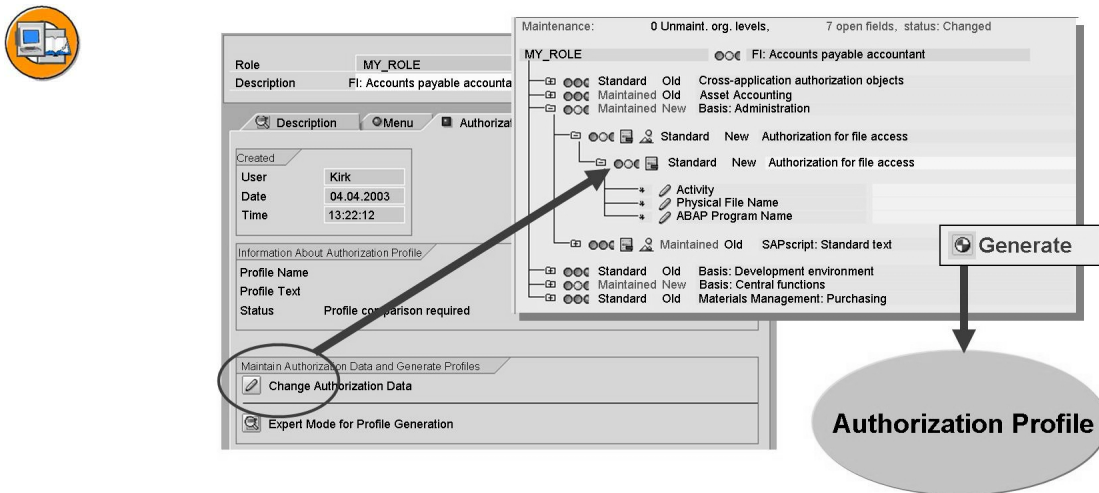


Figure 44: Generating Authorization Profiles

Role maintenance **automatically creates the authorizations** that are associated with the transactions specified in the menu tree. However, all authorization values must be **manually checked and adjusted if required** in accordance with the actual requirements and authorities. The system administrator is responsible for this task, together with the appropriate user department. When using organizational levels, you do not carry out maintenance directly in the field, but by means of the “Org. Levels..” button (Ctrl+F8).

Choose the *Authorizations* tab page and then *Display Authorization Data* or *Change Authorization Data*, depending on the maintenance mode. Check the scope and contents of the authorizations.

If these are proposed by the system, a **green traffic light** in the authorization overview indicates that role maintenance has supplied at least one proposal for each authorization field. A **yellow traffic light** indicates that the authorization must be maintained manually after it has been created. Role maintenance does not provide a default value for the authorization. The above example deals with access to files. Role maintenance cannot “guess” whether data access should only be read access, or should be read and write access.

Some fields appear in many authorizations. A number of important fields were therefore combined into organizational levels, such as the company code. If you maintain an entry for the organizational level using the “Org. Levels..” button, you thus maintain **all** the fields that appear there in one go. A **red traffic light** therefore indicates an unmaintained organizational level.

Once all authorizations are maintained as required, the authorization profile can be generated by choosing *Generate*. Important: The second character of the profile name must not be an underscore (“_”) (see SAP Note 16466). After creation, this name cannot be changed. The authorizations are combined in profiles. The profiles must be entered in the user master record (by the role maintenance) for the authorizations to take effect for the user. This is called *user master comparison*.

Users and Roles

The assignment of users to roles is performed in the role maintenance transaction (transaction PFCG) or in the user maintenance transaction (transaction SU01). Select the tab page *User* and the user IDs to be maintained there. When selecting user IDs, the system uses the current date as the start of the validity period of the assignment; it sets 31.12.9999 as the end date. You can change both values.

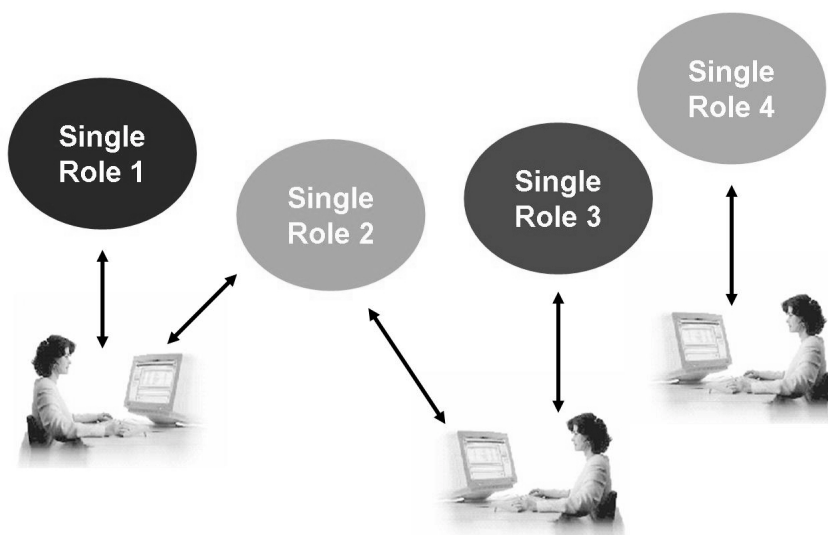


Figure 45: Assigning Roles to Users

Users can be linked with more than one role. This can be useful if some activities (such as printing) are to be permissible across roles.

The assignment of roles to users does not automatically grant the corresponding authorizations to the users. To assign the authorizations, you must first perform a **user master comparison**, during which the role's profiles are entered in the user master record.

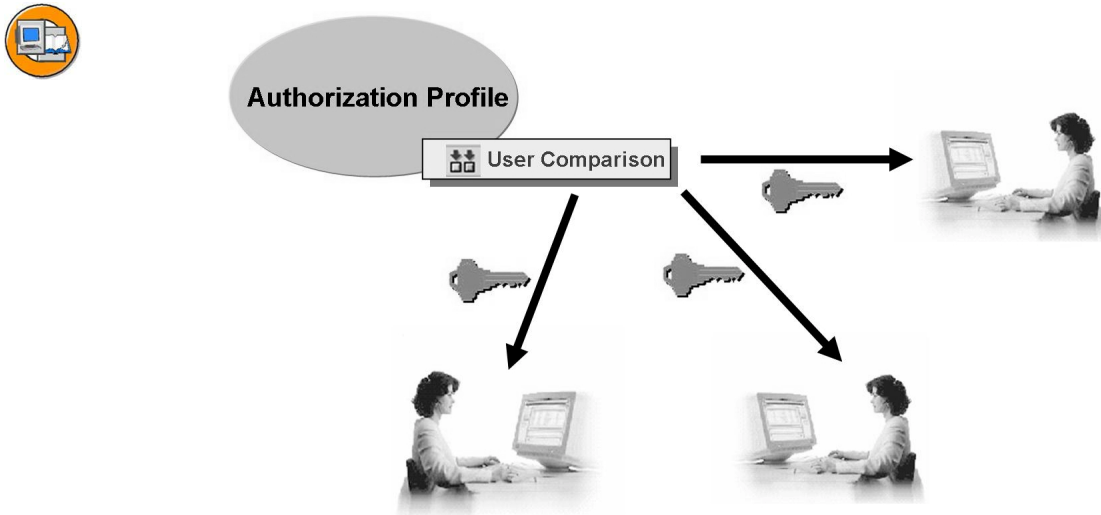


Figure 46: Comparing User Master Records

A user master comparison determines whether authorization profiles should be added to or removed from the current user on the basis of his or her role assignment. During a comparison, profiles are added to a user master due to roles that have been added. If role assignments are manually or time-dependently removed, the corresponding authorization profiles are deleted from the user master record.

The comparison can be performed **for every role individually**. Select the role in role maintenance. Choose the Users tab page and choose User Reconciliation. In the dialog box that the system displays, choose Complete Reconciliation.

If multiple role assignments are to be updated, you can perform a corresponding comparison in role maintenance by choosing *Utilities → Mass comparison* (transaction PFUD). You can individually specify the desired roles, or update all assignments by entering the asterisk (*) character.

You can also activate the periodic user master comparison in role maintenance by choosing *Utilities → Mass comparison*. Choose the option *Schedule or check job for full reconciliation*. The system then displays a search window for the background job *PFCG_TIME_DEPENDENCY*. If it does not find a corresponding job, you can create a new one. The default value is that all user masters are compared once every day.

Exercise 5: Working with Roles

Exercise Objectives

After completing this exercise, you will be able to:

- Create, copy, and modify roles and assign them to users

Business Example

The same scenario as for the last lesson.

Task 1: Copy a Role Template

Copy a role template and assign it to a user.

1. Select the delivered single role SAP_BC_ENDUSER and copy this completely to your own role BC_ENDUSER.
2. Check the transactions assigned for the user menu with this role.
3. Check the authorizations for the role, and maintain open authorizations if necessary.
4. Assign the role to the user ADMIN<##> and save your settings.
5. Perform a user comparison.

Task 2: Create Your Own Role (Optional)

Create your own role.

1. Assign the name MONITORING<##> to the role.
2. Select the transactions SM50, SM51, and SM04 for the role menu.
3. Check the authorizations for the role, and maintain open authorizations if necessary.
4. Assign the role to the user ADMIN<##>.
5. Perform a user comparison.

Task 3: Assign a Role with Transaction SU01

Assign a role to a user in transaction SU01.

1. Check in the user master record ADMIN<##> which roles and profiles are assigned.

Continued on next page

2. Assign the role *ZPFUD* to the user ADMIN<##> using transaction SU01. You can only do this if you are in change mode in the user master record.

Task 4: Check User

Check the user ADMIN<##>.

1. Log on to the SAP System with the user ADMIN<##> and your chosen password and check whether the user can execute the transactions you assigned.

Solution 5: Working with Roles

Task 1: Copy a Role Template

Copy a role template and assign it to a user.

1. Select the delivered single role SAP_BC_ENDUSER and copy this completely to your own role BC_ENDUSER.
 - a) Start transaction PFCG. Place the cursor on the input field for roles. Use the F4 help to select the delivered single role SAP_BC_ENDUSER. Choose the *Copy Role* pushbutton. In the dialog box that appears, enter **BC_ENDUSER** in the *to role* field and choose *Copy all*.
2. Check the transactions assigned for the user menu with this role.
 - a) Choose the *Change* pushbutton or *Role -> Change* from the menu in the initial screen of transaction PFCG for the role BC_ENDUSER. Switch to the *Menu* tab page. Display the transaction code (choose the magnifying glass pushbutton, Switch on technical names) and open the *Basis Functions* folder.
3. Check the authorizations for the role, and maintain open authorizations if necessary.
 - a) Choose the *Authorizations* tab page and then choose *Change Authorization Data*. Check the authorizations for the role and maintain open authorizations if necessary, for example by clicking the yellow traffic light icon at the top and confirming the system query as to whether full authorization should be assigned with *Execute*. *Save* Generate and save your profile settings by choosing *Generate*. Accept the proposed profile name in the process. Leave the *Change Roles: Authorizations* screen by choosing *Back*.



Note: You do not need to *save* again, since this was already performed with the *generate* function.

4. Assign the role to the user ADMIN<##> and save your settings.
 - a) Choose the *User* tab page and enter **ADMIN<##>** in the *User ID* field. *Save* your settings. A user master comparison has not yet been performed, however (next subtask).

If user ADMIN<##> does not exist, create a user with this name in transaction SU01 in a new session.

Continued on next page

5. Perform a user comparison.
 - a) Choose *User Comparison* and then choose *Complete comparison*. Return to the initial screen for PFCG and save all the data that is still required.

Task 2: Create Your Own Role (Optional)

Create your own role.

1. Assign the name MONITORING<##> to the role.
 - a) Start transaction PFCG. Enter **MONITORING<##>** in the input field for roles and choose *Create Single Role*.
2. Select the transactions SM50, SM51, and SM04 for the role menu.
 - a) Choose the *Menu* tab page. Choose the *Transaction* pushbutton and enter the transactions SM50, SM51, and SM04. Then choose *Assign Transactions*.
3. Check the authorizations for the role, and maintain open authorizations if necessary.
 - a) Choose the *Authorizations* tab page and then choose *Change Authorization Data*. Check the authorizations for the role and maintain open authorizations if necessary, for example by clicking the yellow traffic light icon at the top and confirming the system query as to whether full authorization should be assigned with *Execute*. Generate the profile in the same way you did in the previous task and accept the proposed profile name here too. Save your entries and exit the *Change Roles: Authorizations* screen by choosing *Back* (green arrow pointing to the left).
4. Assign the role to the user ADMIN<##>.
 - a) Choose the *User* tab page and enter **ADMIN<##>** in the *User ID* field.
5. Perform a user comparison.
 - a) Choose *User Comparison* and then choose *Complete comparison*.

Continued on next page

Task 3: Assign a Role with Transaction SU01

Assign a role to a user in transaction SU01.

1. Check in the user master record ADMIN<##> which roles and profiles are assigned.
 - a) Start transaction SU01. Enter the name **ADMIN<##>** in the *User* field, and choose the *Change* pushbutton. Choose the *Roles* tab page and check whether the role BC_ENDUSER is entered (this was done in task 1 for this exercise). Choose the *Profiles* tab page, and check that the corresponding profile is entered.
2. Assign the role ZPFUD to the user ADMIN<##> using transaction SU01. You can only do this if you are in change mode in the user master record.
 - a) Choose the *Roles* tab page and enter **ZPFUD** in the *Role* field and confirm with *ENTER*. Save your entries.



Hint: By choosing *Enter* and *Save* in transaction SU01 the user master record comparison was performed.

Task 4: Check User

Check the user ADMIN<##>.

1. Log on to the SAP System with the user ADMIN<##> and your chosen password and check whether the user can execute the transactions you assigned.
 - a) Log on to the SAP system with your ADMIN<##> user. Switch to the user menu and execute some of the assigned transactions.



Lesson Summary

You should now be able to:

- Copy, create, and maintain roles
- Maintain the assignment of roles and users

Lesson: Login Parameters and User Info

Lesson Overview

In this lesson, you learn about important system parameters that are important for user administration, for example, for logon behavior. You can use the information system to obtain information about any incorrect logon attempts. Failed authorization checks are analyzed with the system trace. This lesson will also address using central directory services for maintaining user master record, for example, the address data contained in the records.



Lesson Objectives

After completing this lesson, you will be able to:

- Set system parameters for user logons
- Name standard users in the SAP system
- Locate authorization problems

Business Example

Users are having problems due to missing authorizations. The administrator can analyze these using system tools.

Login Parameters

This lesson deals with authorizations in the SAP system from an administrative point of view. Among other thing, the following questions are considered: Which system settings can be used to influence logon behavior? How can errors and problems be analyzed?



System Profile Parameters	Default	Value Range
Minimum password length <i>login/min_password_lng</i>	6*	1-40 chars *
Validity period for passwords <i>login/password_expiration_time</i>	0	0-1000 days *
Validity period for unused initial passwords <i>login/password_max_idle_initial</i>	0	0-24000 days
Validity period for unused user passwords <i>login/password_max_idle_productive</i>	0	0-24000 days *
Minimum difference in password characters <i>login/min_password_diff</i>	1	1-40 chars *

* New default value and value range since SAP NetWeaver 7.0

Figure 47: System Parameters for User Logons 1/2

You can set the minimum length for passwords with the parameter *login/min_password_lng*. The parameters *login/min_password_digits*, *login/min_password_letters*, *login/min_password_lowercase*, *login/min_password_uppercase*, and *login/min_password_specials* specify the minimum number of **digits**, **letters (number of upper and lower case)** or **special characters** that a password must contain. The value range is 1 to 40.

The parameter *login/password_expiration_time* specifies the number of days after which a user must set a new password. If the parameter is set to 0, the user does not need to change his or her password.

There are general rules for passwords that cannot be deactivated. A password

- Must be at least six characters long (by default)
- Must not begin with “?” or “!”
- Must not be pass
- The new password must differ from the old one by at least 1 character



Hint: The setting that determines that users must create a new password that differs from the previous 5 passwords they have entered is no longer mandatory. You can use the *login/password_history_size* parameter to set the history from between 1 and 100. The proposed standard value remains 5.

You can define additional password restrictions in table *USR40*.

SAP Web Application Server 6.20 and 6.40 offered the parameters *login/password_max_new_valid* and *login/password_max_reset_valid*. They specified for how long an initial password for a newly created user or a password that was reset by an administrator was valid. With SAP NetWeaver AS 7.0, they have been replaced by the parameter *login/password_max_idle_initial*.



Hint: The parameter *login/password_max_idle_initial* indicates the maximum length of time during which an initial password (a password selected by the user administrator) remains valid if it is not used. Once this period has expired, the password can no longer be used for authentication. The user administrator can reactivate the password logon by assigning a new initial password.

Another new parameter that was introduced after SAP Web AS 6.40 is *login/password_max_idle_productive*. This indicates the maximum length of time a productive password (a password chosen by the user) remains valid when it is not used. Once this period has expired, the password can no longer be used for authentication. The user administrator can reactivate the password logon by assigning a new initial password.

With the parameter *login/min_password_diff*, the administrator can determine the number of different characters a new password must possess in comparison with the old one when users change their passwords. This parameter does not take effect when a new user is created or passwords are reset (==> initial password).



System Profile Parameters	Default	Value Range
End the logon procedure <i>login/fails_to_session_end</i>	3	1-99
Maximum number of failed logon attempts <i>login/fails_to_user_lock</i>	5*	1-99*
Deactivation of automatic unlocking <i>login/failed_user_auto_unlock</i>	0*	0-1*
Deactivation of multiple dialog logon <i>login/disable_multi_gui_login</i>	0	0-1
Special users (multiple logon) <i>login/multi_login_users</i>	Alphanumeric	

* New default value and value range since SAP NetWeaver 7.0

Figure 48: System Parameters for User Logons 2/2

You can set the number of failed logon attempts after which SAP GUI is terminated using the parameter *login/fails_to_session_end*. If the user wants to try again, he or she must restart SAP GUI.

You can set the number of failed logon attempts after which a user is locked in the SAP system using the parameter *login/fails_to_user_lock*. The failed logon counter is reset after a successful logon attempt.



Hint: At midnight (server time), the users that were locked as a result of incorrect logon attempts are **no longer automatically** unlocked by the system (default value since SAP NetWeaver 7.0). You reactivate this automatic unlocking with the parameter *login/failed_user_auto_unlock* = 1.

The administrator can unlock, lock, or assign a new password to users in user maintenance (transaction SU01).

If the parameter *login/disable_multi_gui_login* is set to 1, a user cannot log on to a client more than once. This can be desirable for system security reasons. If the parameter is set to 1, the user has the following options when he or she logs on again: Continue with this logon and end any other logons in the system or terminate this logon. Users to whom this should not apply should be specified in the parameter *login/multi_login_users*, separated with commas, and with no spaces.

Initial Passwords for Standard Users



Initial Logon Procedure in SAP Clients

Client	000	001	066	Client (New)
User	SAP*	DDIC	EarlyWatch	SAP*
Initial Password	No longer 06071992 19920706		support	pass



Since these users are public information, they must be protected against unauthorized access. NEW: You are prompted for SAP* and DDIC during the installation in clients 000/001.

Figure 49: Standard Users

Essentially, there are two types of standard users: those created by installing the SAP system and those created when you copy clients.

During the installation of the SAP system, the clients *000* and *066* are created (the client *001* is not always created during an SAP installation; it is also created, for example, during an SAP ECC installation). Standard users are predefined in the clients. Since there are standard names and standard passwords for these users, which are known to other people, you must protect them against unauthorized access.

The SAP system standard user, SAP*

*SAP** is the only user in the SAP system for which no user master record is required, since it is defined in the system code. *SAP** has, by default, the password “*PASS*”, and unrestricted access authorizations for the system.

When you install the SAP system, a user master record is created automatically for *SAP** in client *000* (and in *001* if it exists). At first, this still has the initial password “*06071992*”. The administrator is required to reset the password **during** installation. The installation can continue only after the password has been changed correctly. The master record created here deactivates the special properties of *SAP**, so that only the authorizations and password defined in the user master record now apply.

The DDIC user

This user is responsible for maintaining the ABAP Dictionary and the software logistics.

When you install the SAP system, a user master record is automatically created in client *000* [*001*] for the user *DDIC*. With this user too, you are requested to change the standard password of “*19920706*” during the installation (similar to the user *SAP**). Certain authorizations are predefined in the system code for the *DDIC* user, meaning that it is, for example, the only user that can log on to the SAP system during the installation of a new release.



Caution: To protect the system against unauthorized access, SAP recommends that you assign these users to the user group *SUPER* in the client *000* [*001*]. This user group is only assigned to superusers.

The EarlyWatch user

The EarlyWatch user is delivered in client *066* and is protected with the password “*SUPPORT*”. The EarlyWatch experts at SAP work with this user. This user should not be deleted. Change the password. This user should only be used for EarlyWatch functions (monitoring and performance).



Hint: Special features for the user “*SAP**”

If you copy a client, the user “*SAP**” is always available. This user does not have a user master record, and is programmed into the system code. To protect your system against unauthorized access, you should create a user master record for this standard user. Create a “*superuser*” with full authorization.

If you now delete the user master record “*SAP**”, the initial password “*PASS*” with the following properties becomes valid again:

- The user has full authorization since no authorization checks are made.
- The standard password “*PASS*” cannot be changed.

How can you counter this problem to protect the system against misuse?

- You can deactivate the special properties of *SAP**. To do this, you must set the system profile parameter **login/no_automatic_user_sapstar** to a value greater than zero. If the parameter is active, *SAP** no longer has any special properties. If the user master record *SAP** is deleted, the logon with *PASS* no longer works.
- If you want to reinstate the old behavior of *SAP**, you must first reset the parameter and restart the system.

Determining User Information



Figure 50: Information System

You can call the Information System (transaction SUIM) in the SAP Menu by choosing *Tools* → *Administration* → *User Maintenance* → *Information System* or in user maintenance (transaction SU01) by choosing *Information* → *Information System*.

You can obtain an overview of user master records, authorizations, profiles, roles, change dates, and so on using the information system.

You can display lists that answer very varied questions. For example:

- Which users have been locked in the system by administrators or failed logon attempts?
- When did a user last log on to the system?
- What changes were made in the authorization profile of a user?
- In which roles is a certain transaction contained?

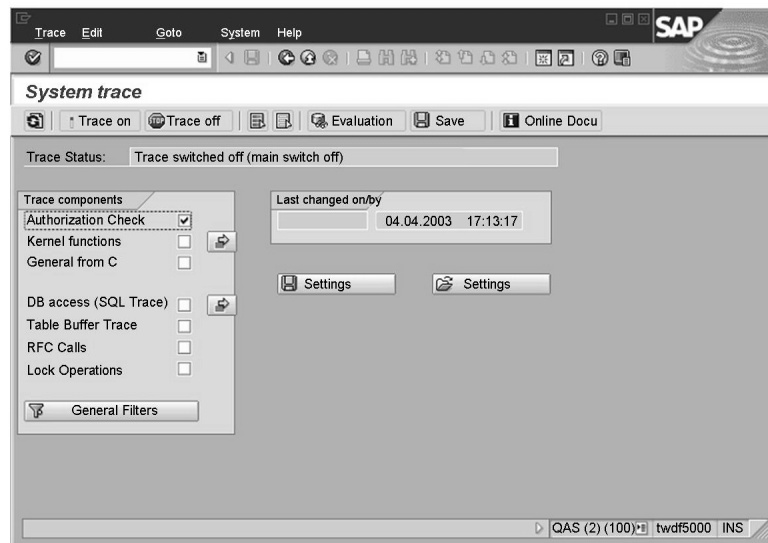


Figure 51: System Trace for Authorizations

You can display the last failed authorization check (transaction SU53) by choosing *System* → *Utilities* → *Display Authorization Check*. The system displays the most recently checked authorization object for which the authorization check was unsuccessful with the checked values.



Hint: Users can only display values for the checked object if they have authorizations for the object *S_USER_AUT*. Otherwise, the text: *No authorization to display authorization values* appears.

The system administrator can use transaction SU53 to check which authorizations were missing for a user for the execution of his or her last (unsuccessful) action. If system administrators have authorizations for *S_USER_AUT* too, they can also display the values that the user has for the checked object.

You can record authorization checks in your own and other sessions using the system trace function *Tools* → *Administration* → *Monitor* → *Traces* → *System Trace* (transaction ST01).



Caution: This only works if the instance (application server) is the same, though.

All checked authorization objects including the checked values are recorded here. The system trace is suited to finding **multiple** missing authorizations. The system trace is activated for the authorization check of a special user who has all required authorizations for the actions to be checked. The actions are performed with this special user. The trace records all authorization checks. These can then be evaluated.



Lesson Summary

You should now be able to:

- Set system parameters for user logons
- Name standard users in the SAP system
- Locate authorization problems

Related Information

SAP Notes

- 2467 - *Password rules & preventing unauthorized logons*
- 862989 - *New password rules as of SAP NetWeaver 2004s (NW AS ABAP 7.0)*

Lesson: Appendix: Advanced User Administration Topics

Lesson Overview

In this lesson, you will obtain an overview of Central User Administration and connections to directory services. These topics are dealt with in detail in SAP course ADM102.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the concept of Central User Administration
- Describe connection to directory services

Business Example

You want to structure the user administration in your company more efficiently by centralizing it.

Central User Administration

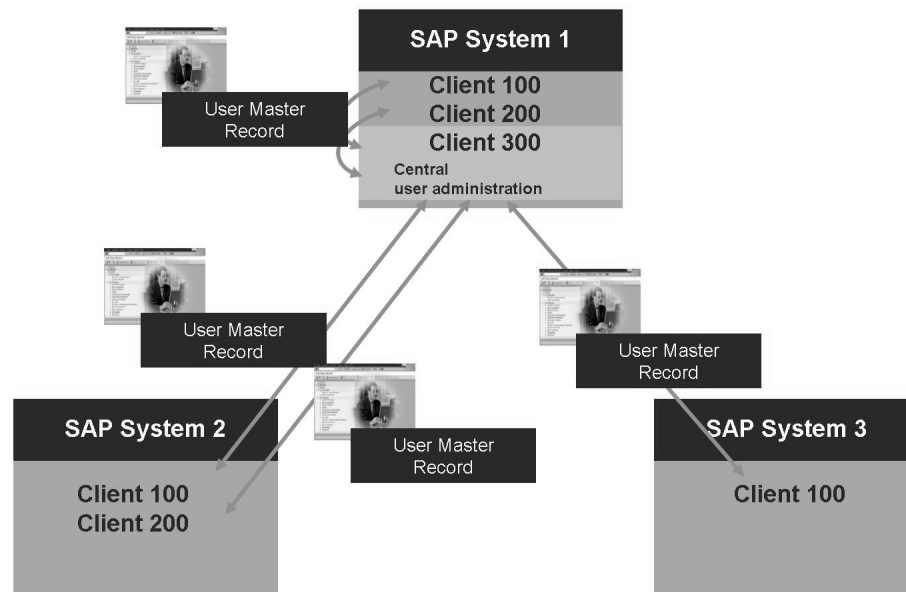


Figure 52: Central User Administration

If you are operating multiple SAP systems with a number of clients, and identical users are created a number of times in different clients, you can significantly reduce your administrative effort for user administration using Central User Administration (CUA). You can perform user maintenance centrally from one client with CUA. This client is then described as the central system. The clients for which user administration is performed from the central system are called child systems.

You can specify for every user which clients it can log on to. Using CUA does not mean that all users can be used in all clients of the system landscape.

You can also specify which user data can only be maintained centrally and which data can also be maintained locally. It is sometimes useful to allow data to be locally maintained by the users or by an administrator. Local maintenance with distribution to all other clients is also possible (for example, in the case of address data being changed).

The **user master data is exchanged** using **ALE**. ALE stands for Application Link Enabling, and is a technology for setting up and operating distributed SAP applications. ALE allows the process-controlled exchange of business messages between loosely connected SAP systems. Asynchronous processing of the communication ensures that application operation is error-free.

Systems that you want to include in a CUA must have at least SAP Basis 4.5.

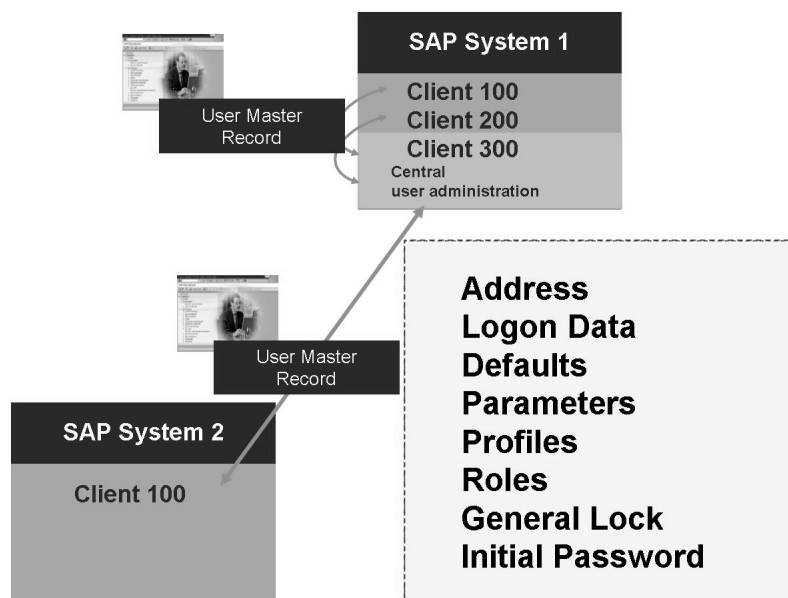


Figure 53: Which Data Can Be Distributed?

The following data can be distributed using Central User Administration (CUA):

- **User master records:** Addresses, logon data, user defaults, and user parameters
- Users are assigned the associated **single and composite roles** and profiles for all child systems. Using CUA has the advantage that you no longer need to log on to each individual client to maintain these assignments locally.
- **Initial password:** When users are newly created, an initial password is transferred to the child systems. This can be changed in the usual way.
- **Lock status:** In addition to the familiar lock reasons (failed logon attempts or locked by an administrator) there is a new general lock. This takes effect in all child systems in which the affected user is permitted and can be removed either centrally or in an individual child system.

You can **assign single or composite roles** and authorization profiles from the central system. However, the authorization profiles are **maintained locally** rather than centrally, since different system settings and release statuses require local administration of authorization profiles.



Note: With central role maintenance, you can define the menu of a role in an SAP system for a different target system. The authorization profiles are always to be maintained in the target system. You can implement the CUA and central role maintenance concepts together or independently.

Central User Administration is discussed in detail in SAP course **ADM102 - SAP Web AS Administration II**.

Directory Services

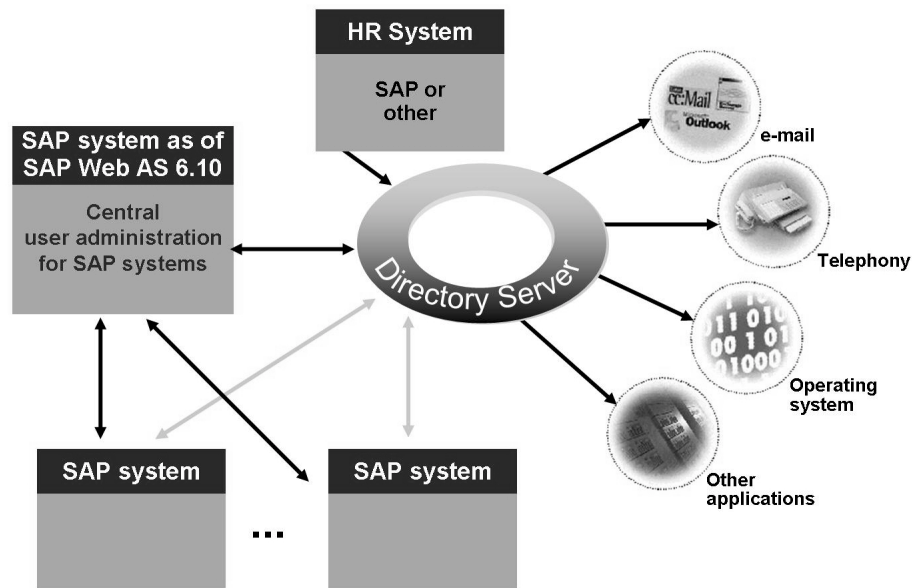


Figure 54: Connection to Directory Services

Directory services allow various applications in an IT landscape to access shared information at a central location. The information is stored on a central directory server that the various systems of your IT landscape can access. In this way, the directory server acts as an “IT address book” for information that is usually used in common, such as personnel data (name, department, organization), user data, and information about system resources and system services. You can use directory services to maintain information in SAP systems for directory-compatible applications (such as user administration or Business Workplace). The standardized Lightweight Directory Access Protocol (LDAP) is usually used as the access protocol. Directory services provide a central information and administration point and therefore simple shared information usage between various applications. Your SAP system can exchange data with directory services using the LDAP protocol. You specify the synchronization direction for each field, that is, whether the SAP system overwrites the data in the directory, or the directory overwrites the data in the SAP system.

The SAP system can exchange data with directory services from various vendors. The SAP system may require attributes that are not in the standard schemata of the directories. SAP usually provides a schema extension for this purpose.



Note: As of SAP Web AS 6.10, SAP systems can easily connect to a directory service. It was possible to connect to a directory service before SAP Web AS 6.10, although rather more effort was involved.



Hint: A connection to a directory service can extend a Central User Administration. That is, these two concepts are **in no way** mutually exclusive, but rather work together very well.

The topic of central directory services/LDAP is dealt with in detail in SAP course **ADM102** - *SAP Web AS Administration II*.



Lesson Summary

You should now be able to:

- Describe the concept of Central User Administration
- Describe connection to directory services

Related Information

- SAP courses
 - ADM940 - *SAP Authorization Concept*



Unit Summary

You should now be able to:

- Create users
- Copy, create, and maintain roles
- Maintain the assignment of roles and users
- Set system parameters for user logons
- Name standard users in the SAP system
- Locate authorization problems
- Describe the concept of Central User Administration
- Describe connection to directory services

Related Information

- SAP courses
 - ADM940 - *SAP Authorization Concept*
 - ADM102 - *Administration AS ABAP II*



Test Your Knowledge

1. How are authorizations assigned to a user?

Choose the correct answer(s).

- ☐ A Users are assigned authorizations using profiles.
- ☐ B Users are assigned authorizations using roles.
- ☐ C Users are assigned authorizations using user names.
- ☐ D Users are assigned authorizations using a Certification Authority.

2. The SAP authorization concept is a positive concept because ...

Choose the correct answer(s).

- ☐ A every user automatically receives all authorizations.
- ☐ B authorizations must be explicitly assigned.
- ☐ C the range of features of the authorization check is so large.
- ☐ D the developers programmed it efficiently.

3. System parameters for the user logon are in the area _____. To display user's incorrect logon attempts, call the Information System with transaction _____. The system trace function is called using transaction _____.

Fill in the blanks to complete the sentence.



Answers

1. How are authorizations assigned to a user?

Answer: A, B

Authorizations are combined into profiles. The roles assigned to users contain profiles with appropriate authorizations for the role. Authorizations are not assigned using user names or a CA.

2. The SAP authorization concept is a positive concept because ...

Answer: B

SAP uses a positive authorization concept. This means that everything that is not explicitly allowed is automatically forbidden.

3. System parameters for the user logon are in the area login. To display user's incorrect logon attempts, call the Information System with transaction SUIM. The system trace function is called using transaction ST01.

Answer: login, SUIM, ST01

User logon settings are implemented using the *login/** parameter. The Information System is called with transaction SUIM, the system trace function with ST01.

Unit 3

User and Authorization Concept AS Java

Unit Overview

This unit covers the user and authorization concept of SAP NetWeaver AS Java or SAP NetWeaver AS ABAP and Java whereby the terms “role” and “group” have **different** meanings than the same terms used in the previous unit. In this unit, however, it is also necessary to use the aforementioned terms and their meanings from the previous unit. If this is the case, they will be known as the “ABAP group”, “ABAP role” or “PFCG role” in this unit.

The structure and configuration of the User Management Engine (UME) and the use of the associated administration tools are explained in this unit. The standard actions in the user administration environment, such as creating users and creating and assigning authorizations and roles (for SAP NetWeaver AS Java), are presented here.



Unit Objectives

After completing this unit, you will be able to:

- List the various UME data sources
- Determine the current data source assignment
- Explain the term UME data partitioning
- Identify and modify configuration parameters
- List and use the tools for administering users and groups
- Explain the terms UME role and J2EE security role
- List the authorization administration tools
- Assign actions to a UME role
- Explain how to assign J2EE security roles to users/groups
- List a number of “special” principles
- Change the password of the administration user
- Activate the emergency user

Unit Contents

Lesson: Architecture and Configuration of the User Management Engine (UME).....	163
Exercise 6: User Management Engine	183
Lesson: User and Group Administration.....	187
Exercise 7: User and Group Administration	195
Lesson: The Java Authorization Concept	203
Exercise 8: The Java Authorization Concept	211
Lesson: Special Principles.....	216
Exercise 9: Default Principles and Emergency Users	223

Lesson: Architecture and Configuration of the User Management Engine (UME)

Lesson Overview

This lesson explains fundamental information about the User Management Engine.



Lesson Objectives

After completing this lesson, you will be able to:

- List the various UME data sources
- Determine the current data source assignment
- Explain the term UME data partitioning
- Identify and modify configuration parameters

Business Example

In your company, AS ABAP and AS Java-based systems are used. You want to ensure consistent user master data within a heterogeneous system landscape.

Fundamentals

AS Java provides an open architecture supported by service providers for the storage of user and group data. The AS Java is supplied with the following service providers which are also referred to as a “user store”:

- DBMS provider: storage in the system database
- UDDI provider: storage via external service providers (Universal Description, Discovery and Integration)
- UME provider: Connection of the integrated User Management Engine

The DBMS and UDDI providers implement standards and therefore ensure that AS Java is J2EE-compliant. When AS Java is installed, SAP's own **User Management Engine (UME)** is always set up as the user store and is the correct choice for most SAP customers. The UME is the only way to flexibly set up and operate user and authorization concepts.

Some of the important features of the UME are:

- The UME has its own administration console for administering users. It allows the administrator to perform the routine tasks of user administration, such as creating users and groups, role assignment, and other actions.
- Security settings can be used to define password policies, such as minimum password length and the number of incorrect logon attempts before a user is locked.
- The UME provides different self-service scenarios that can be used by applications. For example, a user can change his or her data, or register as a new user. Newly-created users can be approved using a workflow.
- User data can be exchanged with other (AS Java or external) systems using an export/import mechanism.
- The UME logs important security events, such as a user's successful logons or incorrect logon attempts, and changes to user data, groups, and roles.

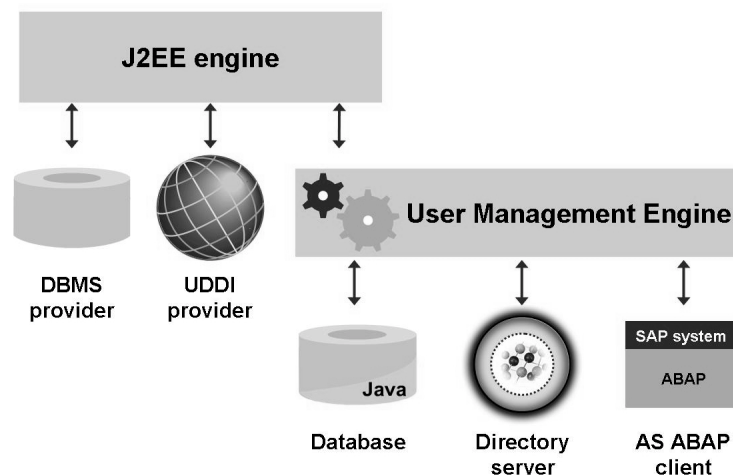


Figure 55: User Store and Data Sources

Architecture

The UME supports a variety of **data sources** where user data can be stored:

- System database
- Directory service (LDAP server)
- ABAP-based SAP system (as of SAP Web AS 6.20)

The figure below shows the architecture of the UME:

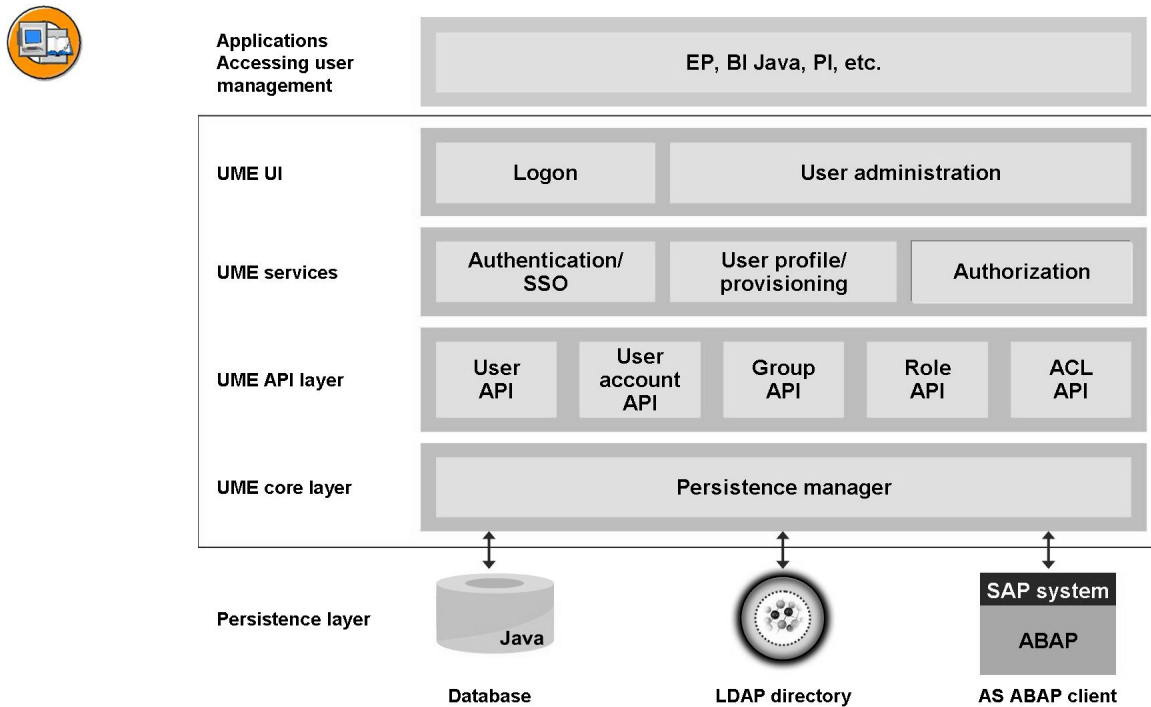


Figure 56: Architecture of the UME:

The UME is a Java application which runs on SAP NetWeaver AS Java and which covers the following functional areas:

- UME Core Layer: Provides persistence managers between the application programming interface and the user management data sources - these control where user data such as users, user accounts, groups, roles and their assignments are read from or written to, with the result that applications which use the API do not have to know where the user management data is stored.
- UME API Layer: This layer provides programming interfaces (APIs) not just for UME developers but also for customers and partners. This means that you can access the UME functions with the Java programs which you develop yourself.
- UME services: The UME provides the following services to higher-level software layers:
 - Log-on procedure and Single Sign-On (log-on to AS Java is taken over for other systems and vice versa)
 - Provisioning processes via user master data
 - Authorization Concept
- UME UI: The UME is responsible for the user interface which, in some log-on procedures, appears in the Web browser, as well as for the UME Administration Console.

The SAP NetWeaver usage types which are based on the AS Java (such as SAP NetWeaver Portal) are based on the UME and perform a number of specific functions on this basis (such as self-registration with approval workflow).

Data Partitioning

As described in the previous section, the UME persistence manager offers the option of storing user data in different data sources. The UME persistence manager also supports data partitioning. This means in practice that, for example, user data for different user types can be stored in different data sources.

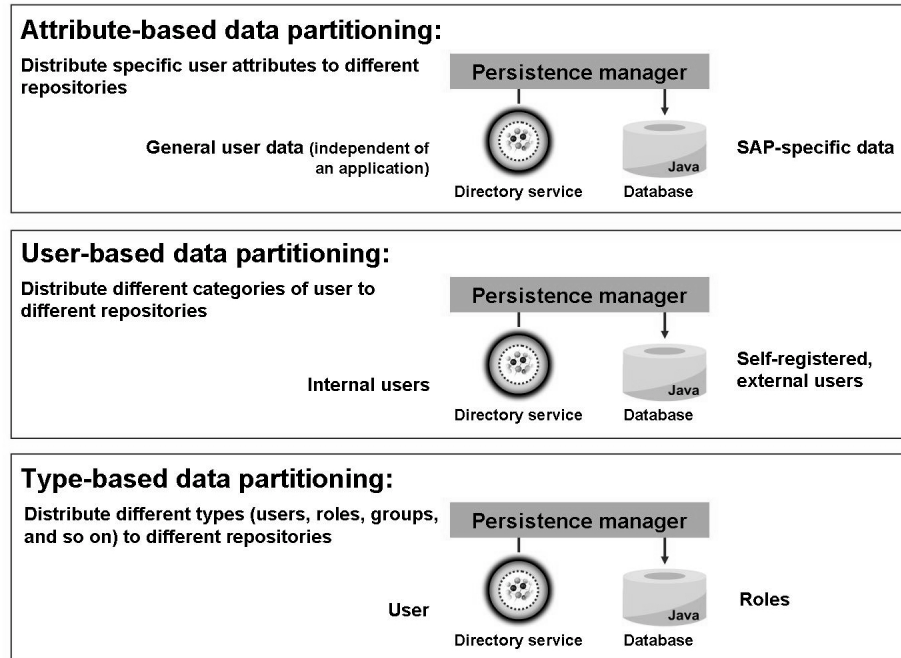


Figure 57: Data Partitioning

In practice, you often work with a combination of the data sources database + directory service or database + ABAP user management. When this is done, certain user attributes are to be stored in a different data source, for example, or users are separated by their categories (internal or self-registered users).

- **Attribute-based data partitioning:** A user in the UME has certain attributes, some of which are classified as global attributes (user ID, telephone number, and so on) and others of which are application-specific. Global information would be particularly suited to being stored in a directory service, and application-specific information in the database.
- **User-based data partitioning:** With this type of partitioning, the data source in which users are stored is decided depending on the category of the user (self-registered or internal users). For example, users that register by self-service can be stored in the database, and internal users in the directory service.
- **Type-based data partitioning:** With type-based data partitioning, different object types can be distributed to different data sources. The types are, for example, users, groups, roles, user accounts. For example, users can be stored in the directory service, and roles in the database.

SAP delivers preconfigured data source combinations (more information will be provided in the next section), which you should only change in special cases. For example, if you are using a directory service as a data source, you may need to perform attribute mapping. You usually use the delivered preconfigured data source combinations without additional changes:

Configuring the Data Source(s)

This section deals with the configuration of the data source(s) stored in the AS Java database in the form of configuration files (in XML format). In most cases, the installation option is retained or the data sources are configured immediately after AS Java installation.

Supported Data Sources and Modification Options

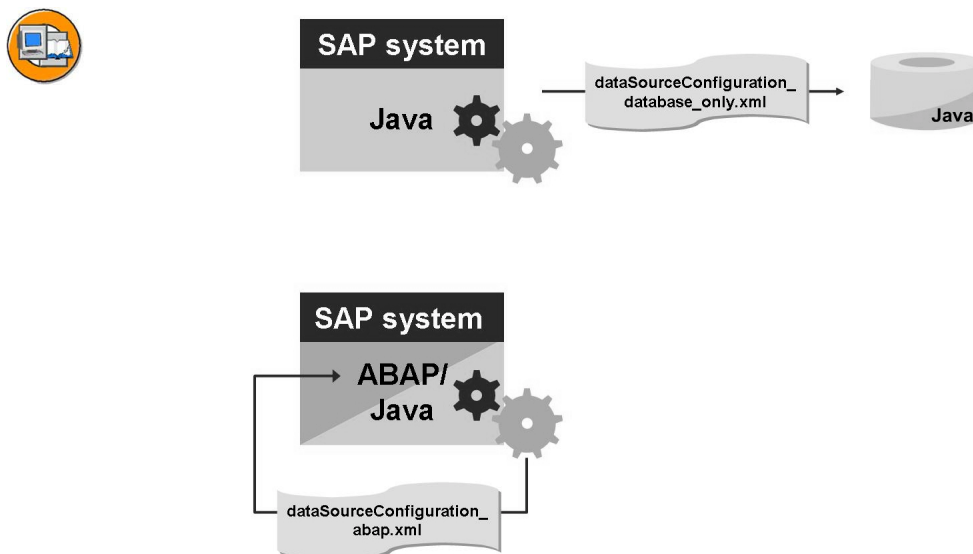


Figure 58: Data Sources after Installation

The data source that is set up during AS Java installation depends on the selected SAP NetWeaver usage type:

- **AS Java (without ABAP):** Data source - system database (configuration file *dataSourceConfiguration_database_only.xml*)
- **AS ABAP +Java:** Data source - ABAP system (configuration file *dataSourceConfiguration_abap.xml*)

Modifying data sources after installation can result in inconsistencies. Restrictions therefore apply to the modification of UME data sources. The following figure explains the supported modification options.



Hint: Please make sure that you observe SAP Note 718383.

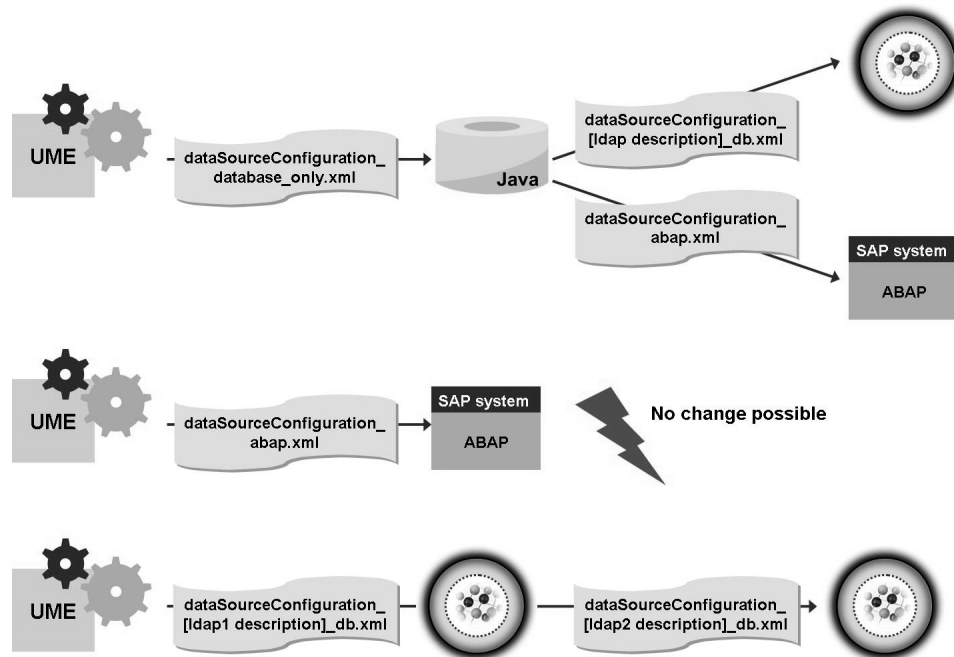


Figure 59: Supported Change Options

The following changes are supported:

- **System database (*dataSourceConfiguration_database_only.xml*):** You can switch to any arbitrary LDAP configuration file (*dataSourceConfiguration_[ldap description]_db.xml*) or an ABAP system (*dataSourceConfiguration_abap.xml*). In this case, you must make sure that the new data source does not contain any users and groups with the same unique attributes as the database (i.e. the new data source must not contain any users or groups with the same unique name or ID as the users or groups in the database).
- **ABAP system (*dataSourceConfiguration_abap.xml*):** No change is possible.
- **Directory service (*dataSourceConfiguration_[ldap description]_db.xml*):** If you have selected an LDAP directory as the user data source, you can modify the structure of the LDAP directory or switch to a different LDAP if this does not modify any unique user IDs.

Below, we present a complex system landscape with AS ABAP, AS Java and non-SAP systems:

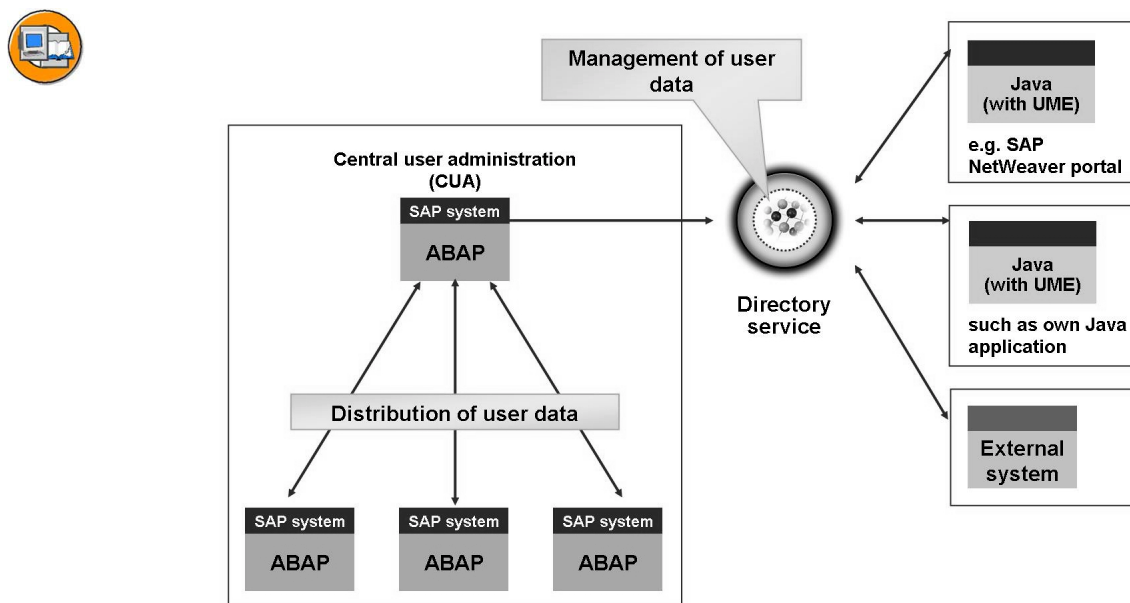


Figure 60: Example of a Heterogeneous System Landscape

In this type of heterogeneous system landscape with SAP systems and non-SAP systems, it is useful to use a directory service as the primary storage location for user data.

As you can see in the figure, the ABAP systems are administered with the central user administration (CUA). The CUA central system synchronizes user data with the directory service. In the case of the AS Java systems, the directory service is configured as the data source. Non-SAP systems also have access to user data through the directory service.

Tools for UME Configuration

The next figure lists the tools with which you can **view and modify** the UME configuration.

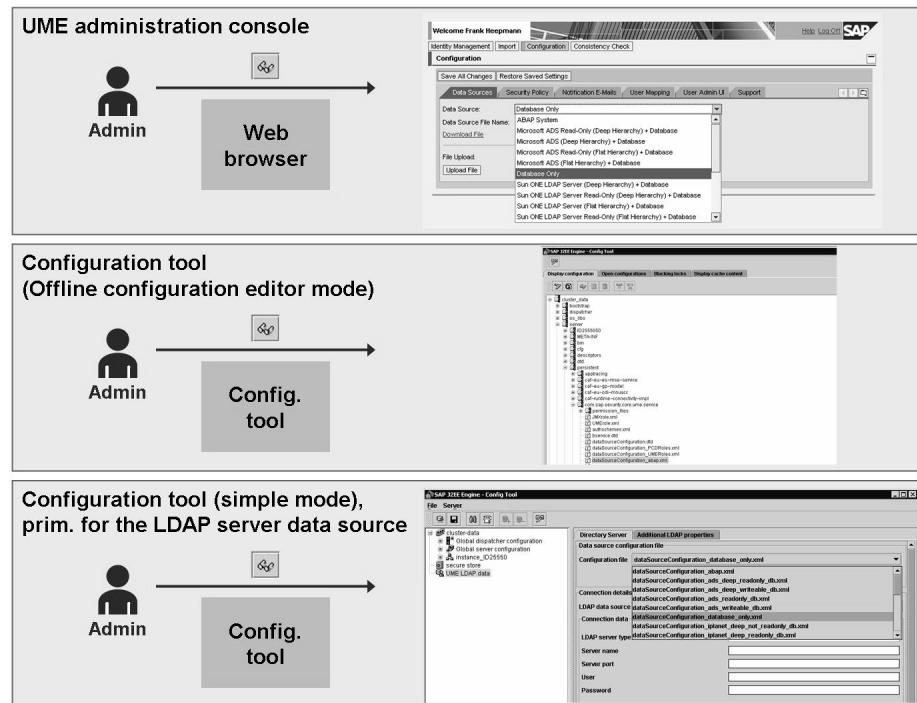


Figure 61: Tools for UME Configuration (Viewing/Modifying)

- **UME Administration Console:** You can use the UME Administration Console running in the web browser to modify selected settings without it being necessary to know the technical parameter names (path: *URL /useradmin → Configuration*).
- **Configuration Tool (Offline Configuration Editor Mode):** Only in Offline Configuration Editor Modus are you able to access all the UME settings (path: *cluster_data → server → cfg → services → PropertySheet com.sap.security.core.ume.service*).
- **Configuration Tool (simple mode):** In the Configuration Tool's simple mode, you will see an area in which you can make settings specially for the LDAP Server data source (path *cluster_data → UME LDAP data*).
- **UME Configuration iView:** If the usage type *EP Core* has been installed in your SAP NetWeaver system, you can use the portal interface to access an iView for UME configuration. This offers similar setting options to the UME Administration Console (path *System Administration → System Configuration → UME Configuration*).



Caution: To apply changes to the UME configuration, it is always necessary to (re)start all the Java instances.

Before you make any changes to the UME configuration, you should first back up the current configuration. You can do this using a function in the UME Administration Console (*Configuration* → *Support* → *Download Configuration Zip File*) which saves the current configuration in a ZIP file. This file allows you to record and trace the changes. However, they are not intended to be re-imported into an AS Java.

The next figure lists the tools with which you can or should only **view** the UME configuration.

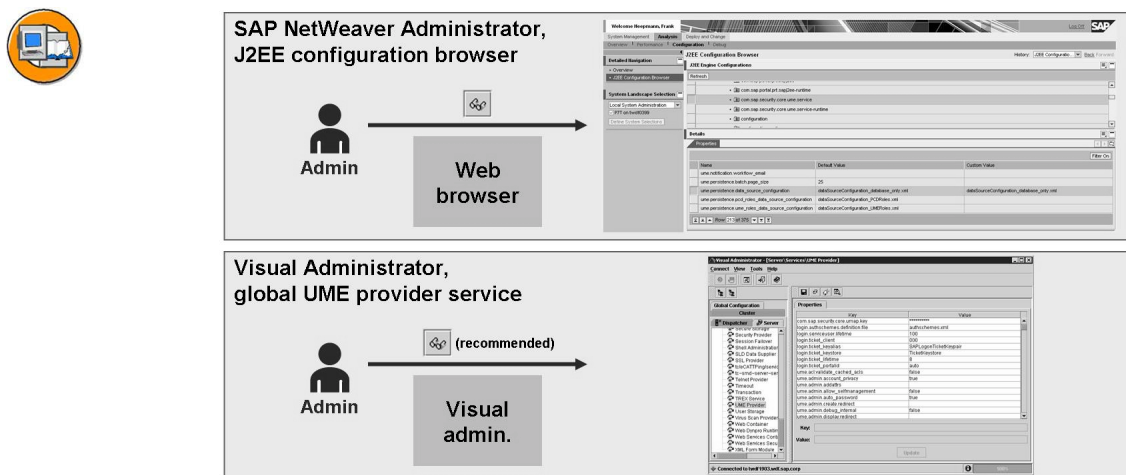


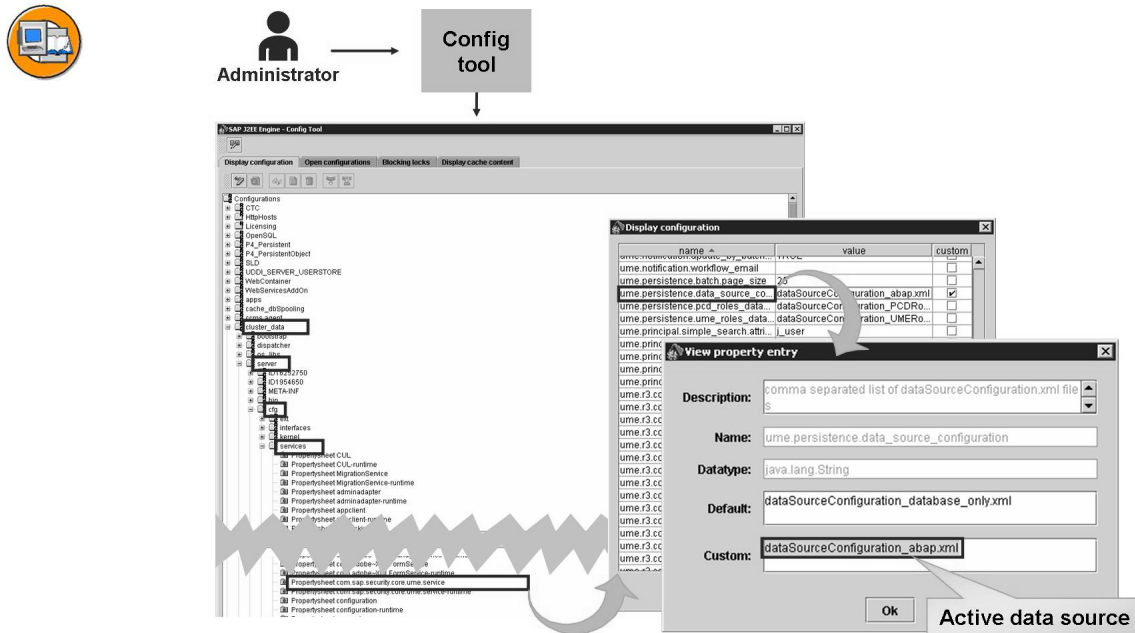
Figure 62: Tools for UME Configuration (View Only)

- **SAP NetWeaver Administrator, J2EE Configuration Browser** You can use the SAP NetWeaver Administrator running in the Web browser to view all the UME parameters (incl. tooltip with descriptive text) (path *Analysis* → *Configuration* → *J2EE Configuration Browser* → <System> → *cluster_data* → *server* → *cfg* → *services* → *com.sap.security.core.ume.service*).
- **Visual Administrator, global UME Provider Service** You should only use the UME Provider Service in the Visual Administrator to view UME parameters (path: *Global Configuration* → *Server* → *UME Provider*).

Since many advanced settings can only be made in Offline Configuration Editor mode, a description of the procedure is presented here:

1. Stop all the Java instances of your system
2. Start the Configuration Tool
3. Switch to Offline Configuration Editor mode
4. Switch to change mode.
5. Navigate to *cluster_data* → *server* → *cfg* → *services* → *PropertySheet* *com.sap.security.core.ume.service*
6. Make the required changes (*Apply Custom*)
7. Start your system's Java instances

By way of an example, the next figure shows how you can find out the currently active data source in Offline Configuration Editor mode.



Path: *cluster_data* >> *server* >> *cfg* >> *services* >> *PropertySheet* *com.sap.security.core.ume.service* >> *ume.persistence.data_source_configuration*

Figure 63: Displaying the Active Data Source

Appendix: Attribute Mapping with Directory Services

As described above, the UME has various preconfigured configuration files in which attribute mapping for directory services can be configured. You can use the Config Tool to view and change these and to configure the attribute mapping.

User data that is sent to a directory service must be appropriately stored in the directory service. Mapping of the attributes is usually necessary to do this. Since different directory services also use different schemas for storing data, you must define which SAP data fields correspond to which directory attributes. If you use the Java API of the user administration component to access user data in your LDAP directory service, you must map the attribute names in the schema of the company's LDAP directory service to the attribute names that are used in the Java API of the user administration component.

This need not always be a one-to-one mapping, but rather one field can be mapped to multiple attributes. The attributes assigned to the fields must also exist in the directory. If not, you need to extend the schema in the directory.

A mapping for the logical attributes of the Java API of the user administration to physical attributes that are used for the *InetOrgPerson* schema in the X.500 standard is delivered in the preconfigured UME XML files. If you use this standard without modifications, you do not need to change the attribute mapping data.

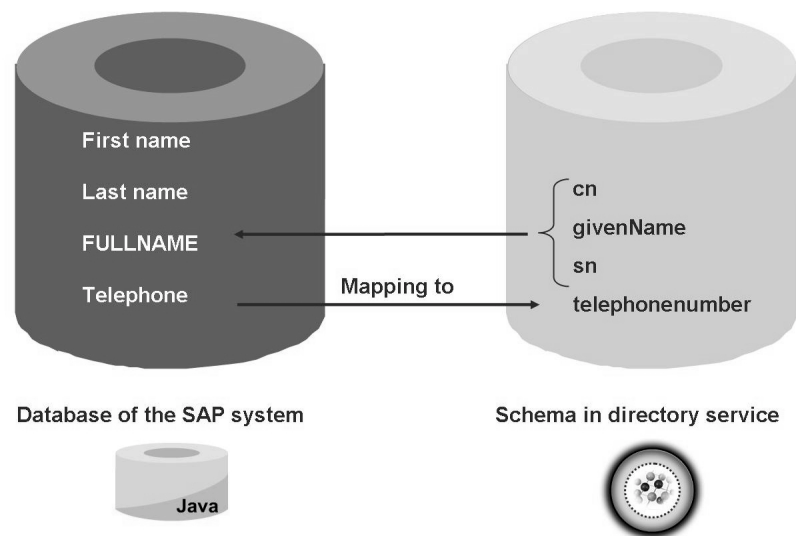


Figure 64: Appendix: Attribute Mapping 1/2

As shown in the figure, the data field FULLNAME (full name) is made up from the attributes *givenName* and *sn* (surname - last name). In the case of the telephone number, for example, the field in the database is *telephone*, while in the LDAP-compatible directory service the field is called *telephoneNumber*.

As described in the previous section, you can use the Config Tool to display the actively used data source and the preconfigured data source combinations as an XML file. The attribute mapping is maintained in the XML configuration file for the data

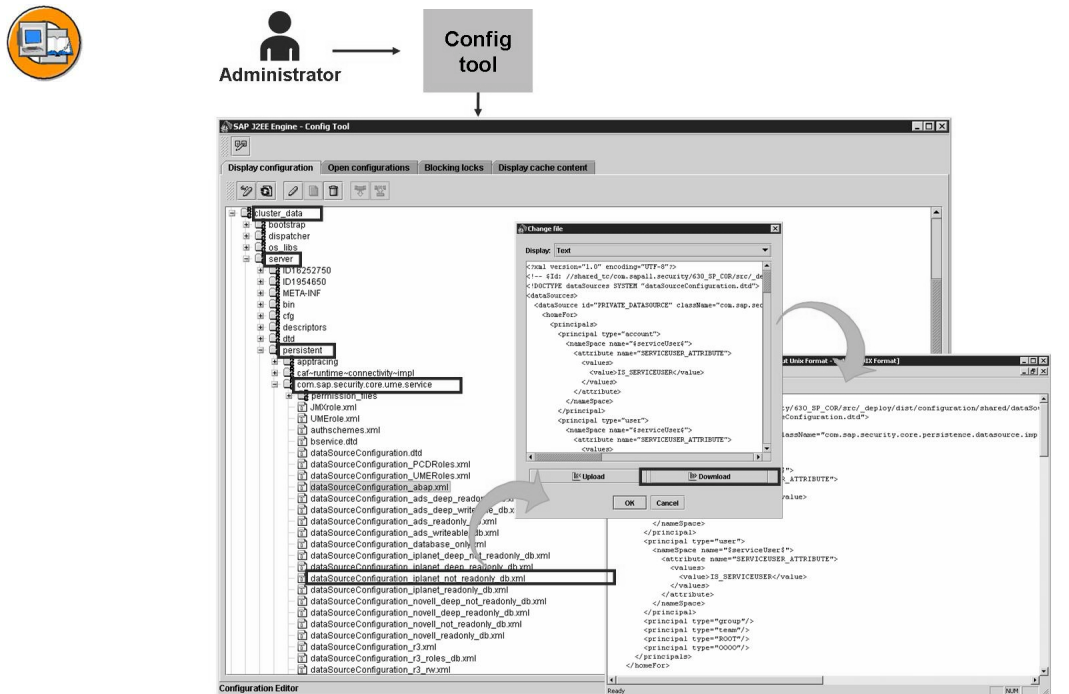
source. You can use a download mechanism in the Config Tool to write the XML configuration files to operating system level, change them there, and then upload them back into the system. You can find the overview of the XML configuration files in the Config Tool:



```
<attributeMapping>
  <principals>
    ...
    <principal type="user">
      <namespaces>
        <namespace name="com.sap.security.core.usermanagement">
          <attributes>
            <attribute name="firstname">
              <physicalAttribute name="givenname" />
            </attribute>
            <attribute name="displayname">
              <physicalAttribute name="displayname" />
            </attribute>
            <attribute name="lastname">
              <physicalAttribute name="sn" />
            </attribute>
            <attribute name="fax">
              <physicalAttribute name="facsimiletelephonenumber" />
            </attribute>
            <attribute name="uniqueusername">
              <physicalAttribute name="uid" />
            </attribute>
            ...
          </attributes>
        </namespace>
      </namespaces>
    </principal>
  </principals>
</attributeMapping>
```

Figure 65: Appendix: XML Files

You can configure the attribute mapping in the relevant XML configuration file. For detailed information about the entire structure of the XML configuration file, see the SAP online documentation. For the attribute mapping, you only need to change the tag `<attributeMapping>` as shown in the figure.



Path: *cluster_data >> server >> persistent >> com.sap.security.core.ume.service >> <file>*

Figure 66: Appendix: Attribute Mapping 2/2

UME Parameters

After you have selected and precisely configured a data source, there are many other parameters with which you can influence the behavior of the UME. The following figure provides an overview of the relevant areas:

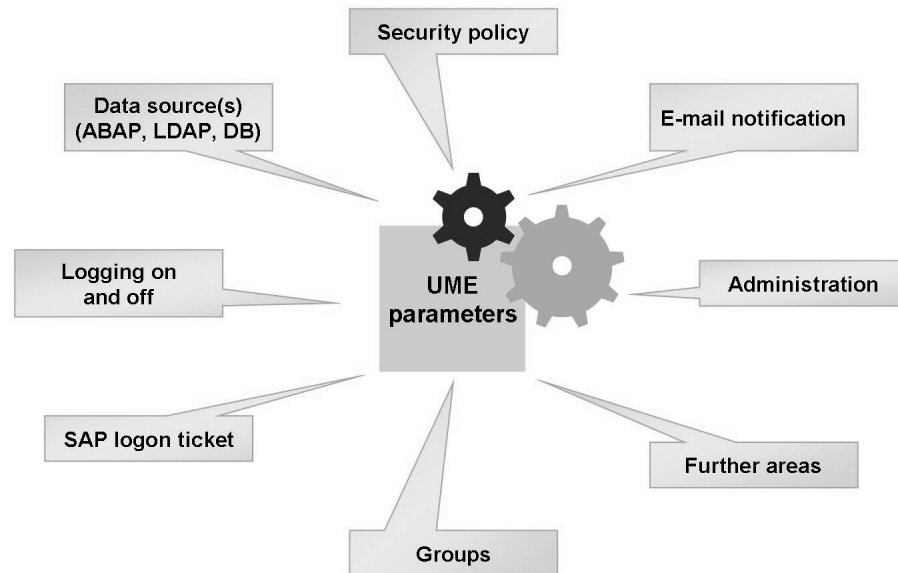


Figure 67: Functions of the UME Parameters:

For a precise description of all the parameters, see the SAP NetWeaver 7.0 online documentation under the path *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security Identity Management* → *Identity Management of the Application Server Java* → *Reference Documentation for Identity Management* → *UME Properties*.

The following list presents a number of important, selected parameters:

Date source(s)

- *ume.persistence.data_source_configuration*:
Name of the UME configuration file (depending on the data source, other parameters may be relevant for connecting the data source)

Security Policy

- *ume.logon.security_policy.auto_unlock_time*
Number of minutes after which a user locked because of invalid login attempts is unlocked again (if the value is 0 then the user remains locked)
- *ume.logon.security_policy.lock_after_invalid_attempts*
Number of invalid login attempts after which a user is locked (automatically set to 0 in an AS ABAP+Java)
- *ume.logon.security_policy.password_special_char_required*
Determines the minimum number of special characters that the password must contain
- *ume.logon.security_policy.password_alpha_numeric_required*
Specifies the minimum number of numeric **and** alphabetical characters that the password must contain (if the number is 3 then the password must contain at least 3 numbers and 3 letters)
- *ume.logon.security_policy.password_expire_days*
Number of days before the password expires
- *ume.logon.security_policy.password_max_length*
or *ume.logon.security_policy.password_min_length*
Maximum or minimum length of the password
- *ume.logon.security_policy.useridmaxlength*
or *ume.logon.security_policy.useridminlength*
Maximum or minimum length of the user ID

E-mail Notification

The UME can be configured in such a way that in certain situations (e.g. after locking a user), e-mails are sent via an external SMTP server. For this to be possible, of course, valid e-mail addresses must be stored in the user master records.

- *ume.notification.mail_host*
Name of the SMTP server for e-mail notification
- *ume.notification.create_performed* or *ume.notification.delete_performed*
An e-mail is sent to the user as soon as the user is created or deleted by the administrator
- *ume.notification.create_approval* or *ume.notification.create_denied*
An e-mail is sent to the user as soon as the administrator approves or rejects the creation of a user account.
- *ume.notification.lock_performed* or *ume.notification.unlock_performed*
An e-mail is sent to the user when the administrator locks or unlocks the user
- *ume.notification.pswd_reset_request*
An e-mail is sent from the user to the administrator when the password is to be reset
- *ume.notification.unlock_request*
An e-mail is sent from the user to the administrator when the account is to be unlocked
- *ume.notification.system_email*
The sender's e-mail address is sent with a dummy name (the address does not have to exist)
- *ume.user_logon_problem_request*
An e-mail is sent from the user to the administrator if the user has logon problems

Logging On and Off

- *ume.logon.branding_image*
Path to the image displayed in the logon screen
- *ume.logon.logon_help*
Activates a "Support" link in the logon screen
- *ume.logoff.redirect.url*
Address that is called following logoff (only for the SAP NetWeaver portal)

SAP Logon Ticket

- *login.ticket_lifetime*
Lifetime of the SAP Logon Ticket (Format <hours>:<minutes>)
- *login.ticket_client*
Dummy “client” written to the SAP Logon Ticket (default 000, in the case of AS ABAP+Java must be set to a client (value) which is not used in the ABAP system)
- *ume.logon.security.relax_domain.level*
Number of subdomains to be removed (a value of 2 means that the SAP Logon Tickets issued by a system on the host *twdf1234.wdf.sap.corp* are sent to servers in the domain *sap.corp*)

Groups

- *ume.supergroups.anonymous_group.uniquename*
ID of the group of anonymous users (default *Anonymous Users*)
- *ume.supergroups.authenticated_group.uniquename*
ID of the group of logged on users (default *Authenticated Users*)
- *ume.supergroups.everyone.uniquename*
ID of the group of all users (default *Everyone*)
- *ume.virtual_groups.names*
IDs of virtual groups (formed on the basis of certain user properties)

Administration

- *ume.admin.addattrs*
Makes it possible to add customer-specific attributes to the user master record
- *ume.admin.search_maxhits*
Maximum number of search hits displayed in the Administration Console (default 1000)
- *ume.admin.search_maxhits_warninglevel*
Number of hits as of which a warning is issued in the Administration Console (default 200)
- *ume.admin.wd.url.help*
URL to the online documentation (may, for example, point to the customer's local help system)
- *ume.admin.wd.table.size.<name>*
Specifies the number of rows for output in the Administration Console (for <name>, there are *small*, *medium* and *large*)

Exercise 6: User Management Engine

Exercise Objectives

After completing this exercise, you will be able to:

- Save UME configuration data
- Determine the current data source
- Modify UME parameters

Business Example

Your company uses SAP NetWeaver Application Server ABAP+Java. Your UME data source consists of a combination of ABAP user management and a database.

Task 1: Configuration Data

Save and evaluate the current configuration data

1. If you have not already done so, log on at your SAP system's operating system level.
2. Log on at the UME Administration Console as user **ADM200-##**.
3. Save the current UME configuration in a file on your SAP server.
4. Using the ZIP file you have just saved, answer the following questions:
 - What data source is currently active?
 - What AS ABAP client is connected?
 - After how many days does the user password expire?
 - For how long are the SAP Logon Tickets issued by the UME valid?

Result

You have saved the current status of the UME configuration in a ZIP file and evaluated it.

Task 2: Modifications

Change a UME setting

1. Use the UME Administration Console to change the threshold value for warnings in the case of extensive search results to **50**.

Solution 6: User Management Engine

Task 1: Configuration Data

Save and evaluate the current configuration data

1. If you have not already done so, log on at your SAP system's operating system level.
 - a) See the task description.
2. Log on at the UME Administration Console as user **ADM200-##**.
 - a) Start a web browser.
 - b) Enter the URL **http://<hostname>.wdf.sap.corp:5<instance>00/useradmin** (example for a QAS group on the host twdf1234: **http://twdf1234.wdf.sap.corp:51000/useradmin**).
 - c) Log on as user **ADM200-##**.
3. Save the current UME configuration in a file on your SAP server.
 - a) Go to the *Configuration* → *Support* view.
 - b) Choose the link *Download Configuration Zip File*.
 - c) Choose *Save* and specify a path on your SAP server.
4. Using the ZIP file you have just saved, answer the following questions:
 - What data source is currently active?
 - What AS ABAP client is connected?
 - After how many days does the user password expire?

Continued on next page

- For how long are the SAP Logon Tickets issued by the UME valid?
- a) In the Windows Explorer, double-click to open the ZIP file which you saved previously.
- b) Double-click to open the contained file: *sapum-global.properties*.
- c) You can use the following UME parameters to answer the questions which are asked:
 - *ume.persistence.data_source_configuration*: Displays the current data source and should be set to *dataSourceConfiguration_abap.xml*
 - *ume.r3.connection.master.client*: Displays the client of the connected ABAP system and should be set to *100*
 - *ume.logon.security_policy.password_expire_days*: Displays the period of validity of passwords in days (in the case of *dataSourceConfiguration_abap.xml*, should correspond to the ABAP parameter *login/password_expiration_time*)
 - *login.ticket_lifetime*: Displays the period of validity of SAP Logon Tickets in hours

Result

You have saved the current status of the UME configuration in a ZIP file and evaluated it.

Task 2: Modifications

Change a UME setting

1. Use the UME Administration Console to change the threshold value for warnings in the case of extensive search results to **50**.
 - a) In the UME Administration Console go to the view *Configuration* → *User Admin UI*.
 - b) Switch to edit mode by choosing *Modify Configuration*.
 - c) Under *Warning Threshold for Large Search Results*, enter **50**.
 - d) Choose *Save All Changes*.
 - e) Use transaction SMICM to restart AS Java instances (*Administration* → *J2EE Cluster (global)* → *Send Hard Shutdown* → *With Restart*).



Lesson Summary

You should now be able to:

- List the various UME data sources
- Determine the current data source assignment
- Explain the term UME data partitioning
- Identify and modify configuration parameters

Related Information

- Online documentation for SAP NetWeaver 7.0: *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security Identity Management* → *Identity Management of the Application Server Java* → *Configuring Identity Management*
- Online documentation for SAP NetWeaver 7.0: *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security Identity Management* → *Identity Management of the Application Server Java* → *Reference Documentation for Identity Management* → *UME Properties*
- SAP Note 718383: *Supported Data Sources and Modification Options*

Lesson: User and Group Administration

Lesson Overview

This lesson presents the tools for the administration of users and groups.



Lesson Objectives

After completing this lesson, you will be able to:

- List and use the tools for administering users and groups

Business Example

You are using AS Java and use a Java application there. To log on to this application, you require a valid user. This must usually first be created. It is also possible to combine multiple users into groups, such as all buyers. Roles (authorizations) are then assigned to the users or groups. Different tools are used, depending on the active data source of the UME.

The Link between Users, Groups and Roles

In the UME environment, the term **Principle** designates the following, central “objects”:



Principles in the UME Environment:

Principle	Meaning
User	General properties of a user (such as name, e-mail, telephone number etc.)
User account	Logon-related properties of a user (such as password, validity, lock indicator etc.)
Group	Set of user and/or groups
Role	Set of (Java) authorizations

For historical reasons, users and user accounts are different principles which are typically associated. When the term *user* is employed below, then, more precisely, it is the associated principles *user* and *user account* that are intended.



Note: Depending on the SAP NetWeaver usage type, the principles have an additional meaning (thus in a SAP NetWeaver Portal there are portal roles that are also handled in the same way as a UME principle).

The following figure shows how you can assign principles.

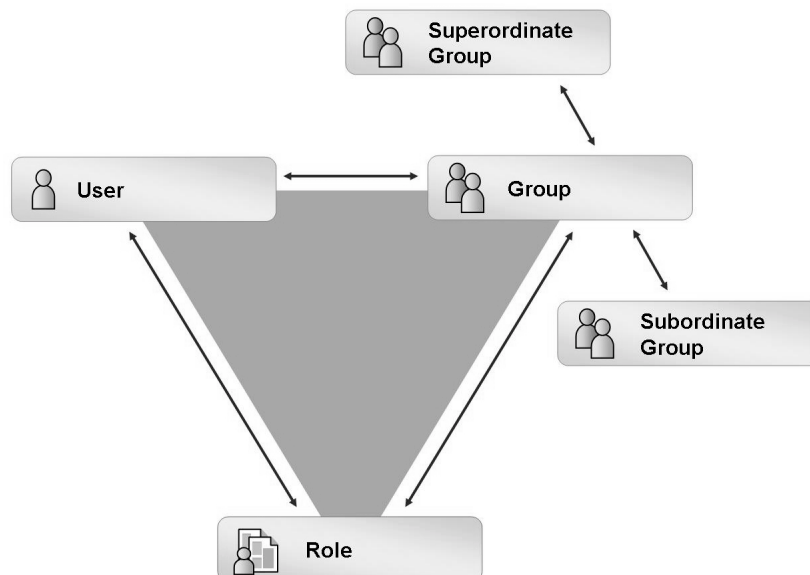


Figure 68: Assigning Principles

Users are usually assigned to groups to which roles are then assigned. However, it is also possible to assign roles to users directly. The Principle group supports hierarchies of groups. A group may also possess superordinate and subordinate groups. Users actually possess the roles which

- are directly assigned to them
- are assigned to the groups to which they belong
- are assigned to the superordinate group of the groups to which they belong

When performing a search in the UME Administration Console, you must check the *Search Recursively* field if you want to see indirectly assigned principles.

Special Features of the ABAP System Data Source

If you use a client of an ABAP system (and consequently the configuration file *dataSourceConfiguration_abap.xml*) as the data source then UME behaves as follows:

- The ABAP users are visible in AS Java and can log onto AS Java with their ABAP passwords.
- The ABAP roles are depicted in AS Java as UME groups of the same name.
- In AS Java, the assignment of ABAP users to ABAP (composite) roles appears as the assignment of UME users to UME groups.

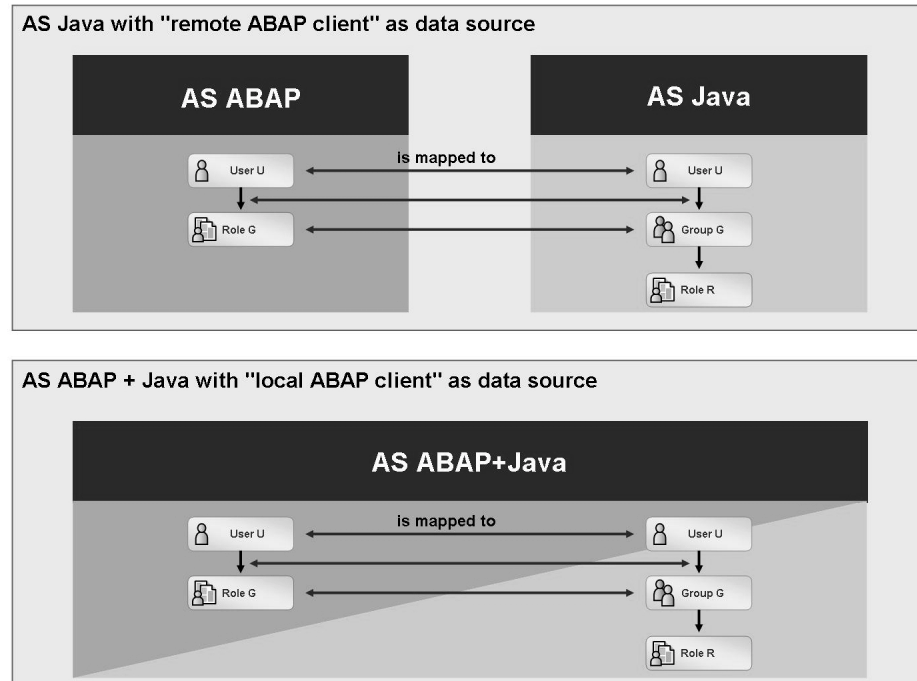


Figure 69: Special Features of the ABAP System Data Source

The reason for this group administration concept is the shared authorization administration for applications that have both ABAP and Java components. Applications such as PI, for example, are made of both ABAP and Java components. The ABAP authorizations are mapped with PFCG roles. The J2EE authorizations are realized using UME roles. A user should be assigned a PFCG role in the ABAP system and a UME role on the Java side for the user to have both ABAP and Java authorizations. To avoid this, the PFCG roles are visible as groups in the UME. The PFCG role (a group) can be assigned a UME role in the UME. If a user is assigned the PFCG role in the ABAP system, he or she automatically also receives the authorizations from the UME role. Assigning authorizations therefore becomes simpler.

The connection between the UME in an AS Java and user management in an AS ABAP is established via the Java Connector (JCo). A communication user existing in ABAP is stored as a UME parameter (this usually has *SAPJSF* in its name). This communication user's ABAP authorization determines whether it is possible to modify ABAP user master records using UME resources.

- The role *SAP_BC_JSF_COMMUNICATION_RO* gives the UME read access to the user data in the AS ABAP.
- The role *SAP_BC_JSF_COMMUNICATION* gives the UME write access to the user data in the AS ABAP.

➔ **Note:** If an ABAP system is used as the data source then certain restrictions apply. These are listed in the online documentation for SAP NetWeaver 7.0.

Administration Tools

The figures in this section explain the tools which you, as administrator, use to maintain users and groups.

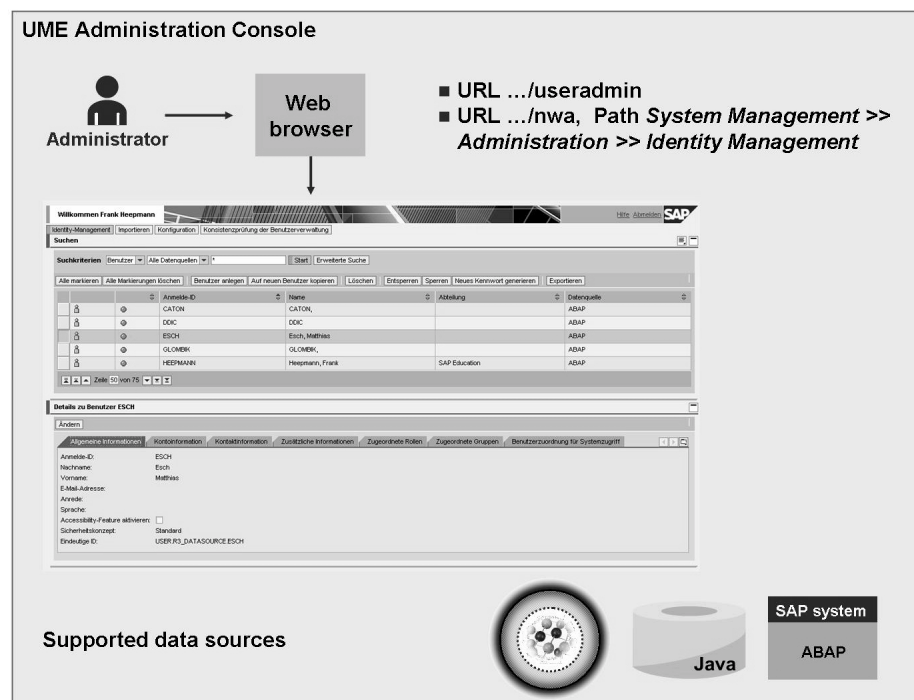


Figure 70: UME Administration Console

The most important tool for a user administrator in an AS Java system is the UME Administration Console. This functions independently of the configured data source and is implemented as an application running in a Web browser (based on Web Dynpro Java). You start the user-friendly Administration Console...

- via the URL *http(s)://<hostname>.<domain>:<http(s) port>/useradmin*
- via the SAP NetWeaver Administrator (URL *.../nwa*) via the path *System Management* → *Administration* → *Identity Management*
- in a portal via the path *User Administration* → *Identity Management*.



Hint: The function scope available in the Administration Console depends on the current user's Java authorizations. For more information, see the lesson “Authorization Concept”.



Visual Administrator

Administrator → Visual administr.

Path: *Server >> Services >> Security Provider >> Runtime >> User Management*

Supported data sources

- Visual Administrator
- Java
- SAP system ABAP

Figure 71: Visual Administrator: User Management

User and group administration in the Visual Administrator is also independent of the configured data source. This application can be accessed via the path *Server <number> → Services → Security Provider → User Management*. However, compared to the Administration Console, only a limited functionality is available to you.

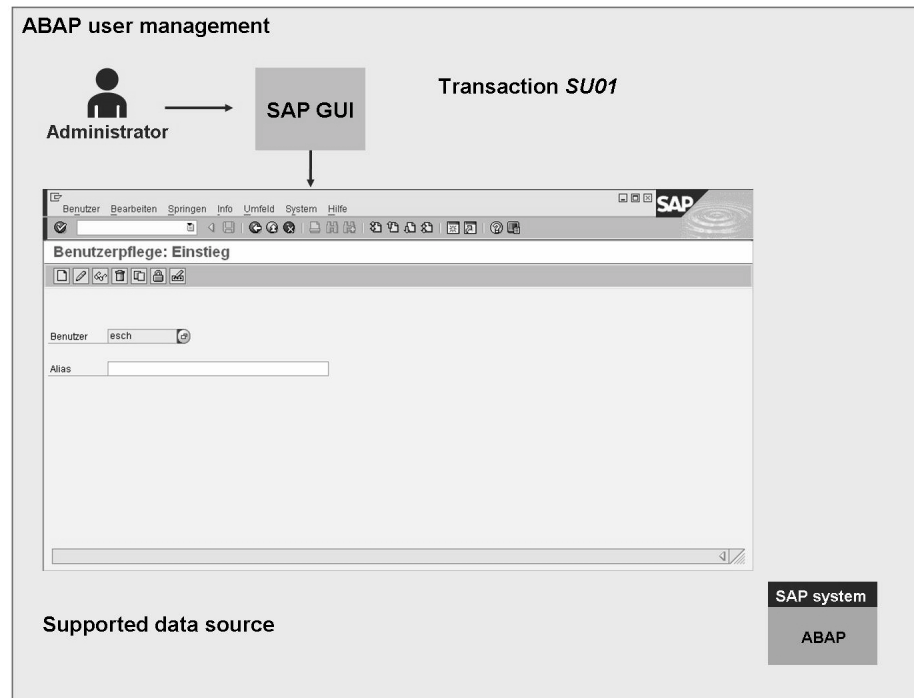


Figure 72: ABAP User Management

If you have used the UME configuration file *dataSourceConfiguration_abap.xml* to connect an ABAP system client, then the usual AS ABAP tools (such as transaction SU01) are available for user administration.

User Types

In the same way as AS ABAP, the UME distinguishes between different **user types** which are listed in the following table:



UME User Types

User Type	Logon to AS Java	Password Rules	Mapped ABAP user types (with ABAP system as data source)
<i>Standard</i>	possible	applies	<i>Dialog</i>
<i>Technical users</i>	possible	does not apply	<i>System</i>
<i>Internal service user</i>	not possible	applies	–
<i>Unknown</i>	possible	applies	<i>Communication, Service and Reference</i>

You specify the user type when you create a user via the **UME Administration Console** (you may not create the type *Unknown*). In the case of existing users, subsequent changes to the user type are only possible with restrictions.

➔ **Note:** The last column in the table is only relevant if you are operating a UME with an ABAP system as the data source. Changes to the user type of an ABAP user are mapped to the corresponding UME user master record (and vice versa if the UME has write access to the ABAP system).

Log and Trace Files

The following log and trace information is particularly relevant in the UME environment

- **Security Log:** File `\usr\sap\<SID>\<instance_number>\j2ee\cluster\server<X>\log\system\security.<n>.log`
- **Security Audit Log:** This is part of the Security Log (category `System//Security/Audit`)
- **Trace Files:** File `\usr\sap\<SID>\<instance_number>\j2ee\cluster\server<X>\log\defaultTrace.<n>.trc`
- **Directory Server Logs:** If you use a directory server as data source, you can monitor the LDAP server accesses and connection pooling

The Security Audit Log allows you to trace changes to principles (e.g. modifications to users or creation of roles). The events that are logged depend on the set weighting. The online documentation for SAP NetWeaver 7.0 describes the weighting associated with each event (path *SAP NetWeaver Library* → *SAP NetWeaver Library by Key Capability* → *Security* → *System Security* → *System Security for AS Java Only* → *Security Audit Log of the AS Java*).

Exercise 7: User and Group Administration

Exercise Objectives

After completing this exercise, you will be able to:

- Administer users and group in the AS Java

Business Example

You are using AS Java and are responsible for user administration. New users should have access to selected applications.

Task 1: User Maintenance

Copy and modify a user using the UME Administration Console

1. Log on at the UME Administration Console as user **ADM200-##**.
2. Copy the template user **JAVATEMPL** to a user **JAVA-##** (## corresponds to your group number).
3. What UME roles does your user **JAVA-##** have?
4. **Optional:** View the user **JAVA-##** in the Visual Administrator.
5. **Optional:** View the user **JAVA-##** in the ABAP User Management.

Result

You can manage users in the UME Administration Console.

Task 2: Group Maintenance

Create and modify UME groups using the UME Administration Console

1. Attempt to start the NWA as user **JAVA-##**.
2. Log on at the UME Administration Console as user **ADM200-##**.
3. Create a UME group named **GROUP-##** and assign the user **JAVA-##** and the UME role **SAP_JAVA_NWADMIN_LOCAL_READONLY** to it.
4. Attempt to start the NWA as user **JAVA-##** again.
5. **Optional:** Display the group **GROUP-##** in the Visual Administrator.
6. **Optional:** Use ABAP tools to assign the user **JAVA-##** the ABAP role **SAP_BC_ENDUSER** and observe the effect in Java.

Continued on next page

Result

You can use the UME Administration Console to manage groups.

Task 3: Change log

Evaluate the Security Audit Log

1. Evaluate the most recent entries in the Security Audit Log (using a tool of your choice).

Solution 7: User and Group Administration

Task 1: User Maintenance

Copy and modify a user using the UME Administration Console

1. Log on at the UME Administration Console as user **ADM200-##**.
 - a) Start a web browser.
 - b) Enter the URL **http://<hostname>.wdf.sap.corp:5<instance>00/useradmin** (example for a QAS group on the host twdf1234: **http://twdf1234.wdf.sap.corp:51000/useradmin**).

Note: Alternatively, you can launch the UME Administration Console via the NWA.
 - c) Log on as user **ADM200-##**.
2. Copy the template user **JAVATEMPL** to a user **JAVA-##** (## corresponds to your group number).
 - a) In the Administration Console's *Identity Management* area, run a search for the user **JAVATEMPL**.
 - b) Select the hit **JAVATEMPL** and choose *Copy to New User*.
 - c) In the *General Information* tab, enter the *Logon ID* (set to **JAVA-##**), *Password* and *Last Name* (any).
 - d) Leave the other fields unchanged and *Save* the data.
3. What UME roles does your user **JAVA-##** have?
 - a) In the Administration Console, view the details for the user **JAVA-##** in display mode.
 - b) Go to the *Assigned Roles* tab. Check *Search Recursively* and choose *Start*.

You should see that the copied user has the same roles and (groups) as the copy template.

Continued on next page

4. **Optional:** View the user **JAVA-##** in the Visual Administrator.
 - a) If you have not already done so, log on as the user **ADM200-##** at your system's Visual Administrator.
 - b) Navigate to the entry *Server <number> → Services → Security Provider → Runtime → User Management*.
 - c) In the field *Users / Name* field, enter ***##** (## again stands for the group number) and choose *Search*.

In the result list, you should see, among other things, the user **JAVA-##**.
5. **Optional:** View the user **JAVA-##** in the ABAP User Management.
 - a) If you have not already done so, log on as the user **ADM200-##** at your system's SAP GUI.
 - b) Start transaction SU01.
 - c) In the field *User* field, enter ***##** (## again stands for the group number) and choose the F4 search.

In the result list, you should see, among other things, the user **JAVA-##**.

Result

You can manage users in the UME Administration Console.

Task 2: Group Maintenance

Create and modify UME groups using the UME Administration Console

1. Attempt to start the NWA as user **JAVA-##**.
 - a) Close any Web browser windows.
 - b) Enter the URL **http://<hostname>.wdf.sap.corp:5<instance>00/nwa** (example for a QAS group on the host twdf1234: **http://twdf1234.wdf.sap.corp:51000/nwa**).
 - c) Enter the logon data for the user **JAVA-##** (and change the password).

The will see a message informing you that you do not have the necessary authorizations.

Continued on next page

2. Log on at the UME Administration Console as user **ADM200-##**.
 - a) Enter the URL **`http://<hostname>.wdf.sap.corp:5<instance>00/useradmin`** (example for a QAS group on the host twdf1234: **`http://twdf1234.wdf.sap.corp:51000/useradmin`**).**Note:** Alternatively, you can launch the UME Administration Console via the NWA.
 - b) Log on as user **ADM200-##**.
3. Create a UME group named **GROUP-##** and assign the user **JAVA-##** and the UME role **SAP_JAVA_NWADMIN_LOCAL_READONLY** to it.
 - a) In the Administration Console's *Identity Management* area, select the *Groups* view.
 - b) Activate *Create Group*.
 - c) In the *General Information* tab, enter **GROUP-##** under *Unique Name*.
 - d) Go to the *Assigned Users* tab. Under *Available Users* search for the user **JAVA-##**. Select this entry and click *Add*.
 - e) Go to the *Assigned Roles* tab. Under *Available Roles* search for the role **SAP_JAVA_NWADMIN_LOCAL_READONLY**. Select this entry and click *Add*.
 - f) *Save* the group.
4. Attempt to start the NWA as user **JAVA-##** again.
 - a) Close any Web browser windows.
 - b) Enter the URL **`http://<hostname>.wdf.sap.corp:5<instance>00/nwa`** (example for a QAS group on the host twdf1234: **`http://twdf1234.wdf.sap.corp:51000/nwa`**).
 - c) Enter the logon data for the user **JAVA-##**.

You can now work with the NWA (locally, for viewing).

Continued on next page

5. **Optional:** Display the group **GROUP-##** in the Visual Administrator.
 - a) If you have not already done so, log on as the user **ADM200-##** at your system's Visual Administrator.
 - b) Navigate to the entry *Server <number> → Services → Security Provider → Runtime → User Management*.
 - c) In the field *Groups / Name* field, enter ***##** (## again stands for the group number) and choose *Search*.

In the result list, you should see, among other things, the group **GROUP-##**.
6. **Optional:** Use ABAP tools to assign the user **JAVA-##** the ABAP role **SAP_BC_ENDUSER** and observe the effect in Java.
 - a) If you have not already done so, log on as the user **ADM200-##** at your system's SAP GUI.
 - b) Start transaction SU01.
 - c) In the *Users* field, enter **JAVA-##** and choose *Change*.
 - d) Enter the ABAP role **SAP_BC_ENDUSER** in the *Roles* tab and *Save*.
 - e) Call the Administration Console as user **ADM200-##**.
 - f) View the users assigned to the user **JAVA-##**.

The user **JAVA-##** should be assigned (via the data source “R3_ROLE_DS”) to the **group** **SAP_BC_ENDUSER** in Java. This group could now be used to assign authorizations in Java via UME roles.

Result

You can use the UME Administration Console to manage groups.

Continued on next page

Task 3: Change log

Evaluate the Security Audit Log

1. Evaluate the most recent entries in the Security Audit Log (using a tool of your choice).
 - a) Start (as user **ADM200-##**) a log evaluation tool (e.g. the NWA or the Visual Administrator).
 - b) Open the files `\usr\sap\<SID>\<instance_number>\j2ee\cluster\server<X>\log\system\security.<n>.log` for all the server processes in your Java cluster.
 - c) Set a filter for the category *System/Security/Audit*.

The displayed entries allow you to identify who performed what operation and when.



Lesson Summary

You should now be able to:

- List and use the tools for administering users and groups

Related Information

- Online documentation for SAP NetWeaver 7.0: *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security* → *Identity Management* → *Identity Management of the Application Server Java* → *Administration of Users and Roles*

Lesson: The Java Authorization Concept

Lesson Overview

To access an application, authentication is usually required. Not all users perform the same actions. Authorizations control which functions are permitted for a user. These authorizations must be assigned to a user.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the terms UME role and J2EE security role
- List the authorization administration tools
- Assign actions to a UME role
- Explain how to assign J2EE security roles to users/groups

Business Example

SAP systems perform authorization checks within the SAP NetWeaver platform with a role-based identity management approach. This means that you assign authorizations to users or groups for a specific system on the basis of the tasks that they are to perform.

Users and Authorizations in SAP NetWeaver AS Java

You can use authorizations to control which users can access a Java applications, and which users are permitted for a user. Authorizations are combined as roles and then assigned to a user or a user group by an administrator. The UME administration console and Visual Administrator tools are used to assign authorizations.

Authorization checks are built into a Java application. You must distinguish between the following authorization checks:



- J2EE security roles
- UME roles



Caution: As a matter of principle, UME roles can only be administered using the UME Administration Console, and J2EE security roles can only be administered using the Visual Administrator.

With both types of authorization check, the developer needs to define the authorizations query in the application. The developer decides which type of authorization check is to be used. This means in practice that whether J2EE security roles or UME roles are used depends on the application.

J2EE security roles are part of the J2EE standard. UME roles are an (SAP) extension of the J2EE security roles. You can define the same authorization checks with J2EE security roles and UME roles. However, it is easier and more precise to assign authorizations with UME roles. A J2EE security role comprises one object and UME roles many authorization objects (known as **actions**). This means that many J2EE security roles but perhaps only one UME role need to be assigned for the same authorizations. We recommend that you always use UME roles, except in cases in which J2EE security roles are sufficient.



Note: A role in the ABAP environment is roughly equivalent to a UME role. An authorization object in the ABAP environment can be compared to a security role.

Appendix: Declarative and Programmable Authorizations

Authorizations can be defined as either declarative or programmable:

- Declarative security means that the Java container forces the access control, without the programming work by the developer being required. In the container of an application, an additional object (role) is defined, which is added to the objects of an application. Before each call, the application checks whether the user has authorization for this application.
- Programmable security means that the developer uses a method to check whether a caller of an EJB or a Web resource has a specific role. The authorization check is defined directly in the coding as source code. The developer can use these “role references” to control the display of individual control elements. This means, for example, that users to whom the queried role is assigned can receive a more comprehensive display on the same Web page than users to whom this role is not assigned.

J2EE security roles are used for declarative authorization checks and UME roles for the programmable approach.

J2EE Security Roles

J2EE security roles are part of the J2EE standard.

A security role is an abstract logical definition that protects access to an application, a service, or another resource. The consists of only a name and a description. The role relates only to the application for which it was defined.

J2EE security roles allow an access check for J2EE applications. The authorizations are defined declaratively. A developer creates a J2EE security role for each new application object. These objects are consolidated during the assembly process and made available on the J2EE server. A user can use these objects only if the administrator has specified the user or group name in the J2EE security role.

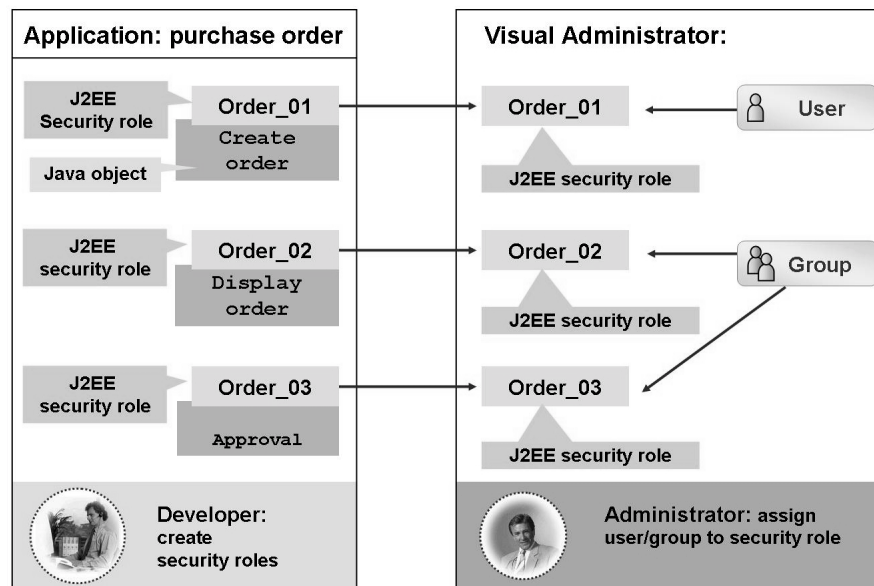


Figure 73: Structure of J2EE Security Roles

The figure shows the *Order* application as an example. For this application, a developer creates objects such as *Create order*, *Approve order*, and so on. If you are using J2EE security roles, a security role must be created for each object. The role is defined in the deployment descriptor (XML file) of a specific application. If the application is made available on the J2EE server, the administrator must add user names or user groups to each of these security roles for the users that are to use this application. The administrator must assign each single authorization/J2EE security role individually to a user or a group.

This authorization concept is suitable for small J2EE applications. It can protect resources such as URLs or EJB methods. It is also possible to use roles to protect resources defined by a service (keystore view). These roles can be created automatically by the service or manually by the administrator.



Note: The security role *administrators* is assigned to the Administrators group. This means that only members of the Administrators group can access applications or resources that are protected with the security role *administrators*.

Assigning a J2EE Security Role

You can use the Visual Administrator to assign security role to a user or group. The Security Provider service of SAP NetWeaver AS Java must be running, and the user that wants to make the assignment must have administration authorizations. A J2EE security role can be assigned

- either directly to users and/or groups
- or as a so-called reference role to precisely one J2EE security role in the component *SAP-J2EE-Engine*

To assign security roles, proceed as follows:

1. Start the Visual Administrator (`\usr\sap\<SID>\<instance>\j2ee\admin\go`).
2. Navigate to *Server* → *Services* → *Security Provider* → *Runtime* → *Policy Configurations*.
3. In the *Components* area, select the application (or service).
4. Choose the *Security Roles* tab page.
5. In the *Security Roles* area, select the security role that you want to assign.
6. Switch to change mode if necessary.
7. Depending on the type of J2EE security role, you either
 - perform assignment directly to users and/or groups
 - perform assignment to a reference security role



Figure 74: Assigning a J2EE Security Role (Visual Administrator)

UME Roles

In the UME, there is a role concept with which authorizations are assigned. These authorizations relate to authorization checks that are defined in the coding of the SAP Java application. The authorization concept in the UME uses permissions, actions, and roles.

Permissions are defined in the Java coding. This is known as programmable security. Permissions are used to provide an access control. Permissions cannot be assigned directly to a user.

An **action** is a collection of permissions. A Java application defines its own actions and specifies the authorizations in an XML file *<name of the application>.xml* (e.g. *sap.com_TC~wd~dispwda.xml*). Actions are displayed in the UME Administration Console. You can use the UME Administration Console to combine these actions into **roles**.

UME roles group actions of one or more applications. You can assign UME roles to users in the UME Administration Console.

SAP's Java applications work with UME roles. If SAP delivers a Web Dynpro application, you can only assign authorizations using UME roles.

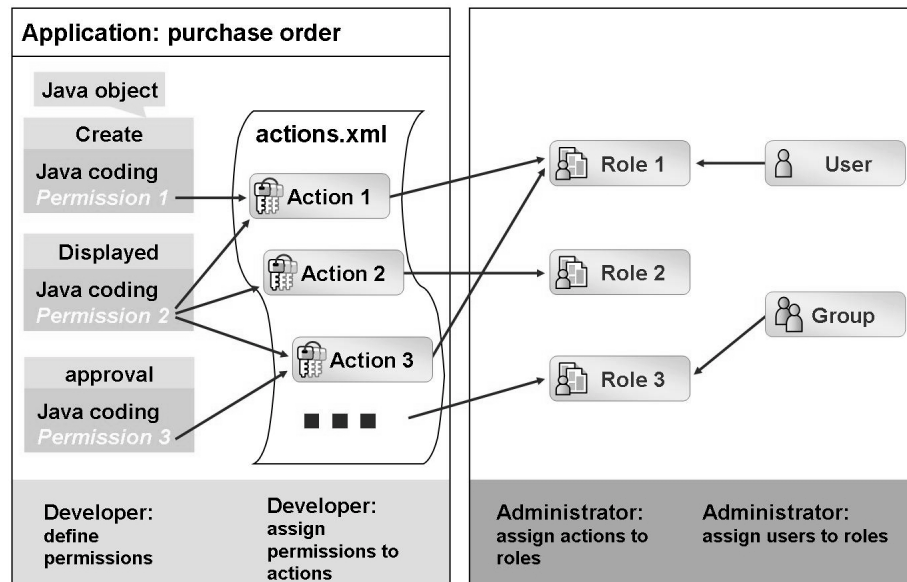


Figure 75: Structure of UME Roles

The figure shows the *Order* application as an example. This application consists of multiple objects, such as *Create order*, *Approve order*, into which a developer has built the corresponding authorization check directly in the coding. With UME roles, permissions (authorization objects) are defined directly in the coding and then bundled into actions by the developer. The administrator can then combine these actions into roles, and assign them to users.

Using this concept, developers can define very detailed authorizations even though the complexity resides in just a few actions. Actions are predefined by the developer, delivered to customers together with the application, and available as an XML file. This allows a simple and clear authorization concept for large Java applications.

Assigning UME Roles

You can use the UME administration console to maintain UME roles. You perform both the assignment of actions to UME roles and the assignment of roles to UME users or groups there.

After logging on with an administrator user, select the appropriate role, display the assigned actions, and change the role, if necessary. Then assign the role to a user and/or a group.



Figure 76: Maintaining UME Roles (UME Administration Console)

It is particularly important for the administration of authorizations that the Java application itself provides UME with a large number of actions. These UME actions permit the precise definition of the rights which users have to principles (e.g. “display all users” or “maintain all groups”). The UME actions supplied by SAP are described in the online documentation (path: *SAP NetWeaver Library → SAP NetWeaver by Key Capability → Security Identity Management → Identity Management of the Application Server Java → Reference Documentation for Identity Management → Standard UME Actions*).

You can adjust the authorizations delivered by SAP. To do this follow the description in the online documentation under *SAP NetWeaver Library → SAP NetWeaver by Key Capability → Security Identity Management → Identity Management of the Application Server Java → Reference Documentation for Identity Management → Developer Documentation for Identity Management*).

The following, final figure illustrates analogies between the authorization concepts in AS ABAP, AS Java and the J2EE standard:

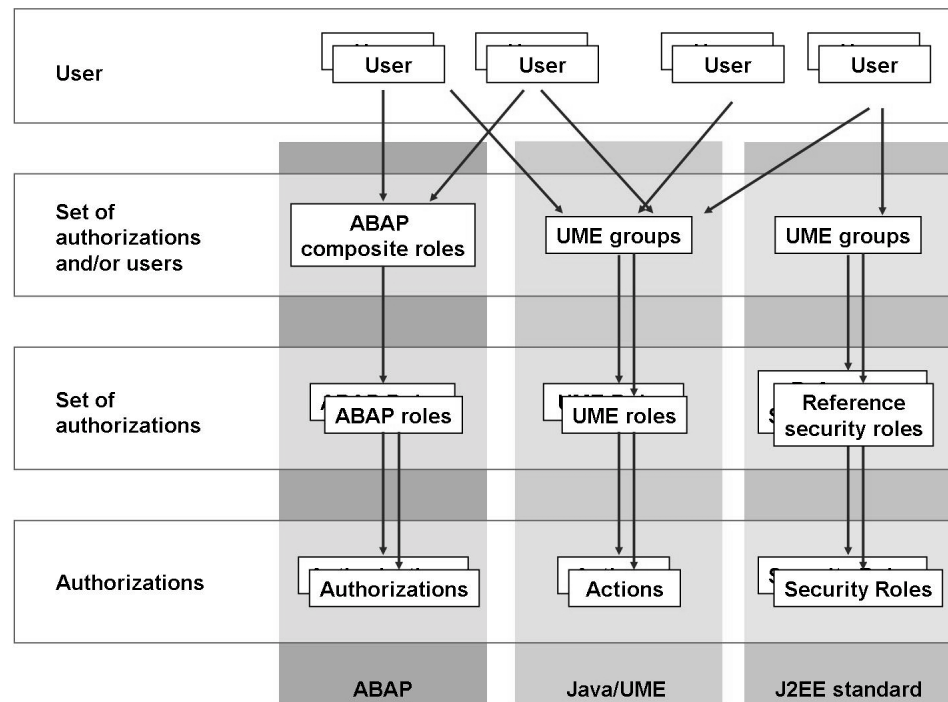


Figure 77: Comparison of the Authorization Concepts

Exercise 8: The Java Authorization Concept

Exercise Objectives

After completing this exercise, you will be able to:

- Analyze J2EE security roles
- Enter actions in UME roles and assign to users/groups

Business Example

SAP systems perform authorization checks within the SAP NetWeaver platform with a role-based identity management approach. This means that you assign authorizations to users for a specific system on the basis of the tasks that they are to perform.

Task 1: J2EE Security Roles

Evaluate the assignment of J2EE security roles

1. Can user **JAVA-##** log onto your AS Java via Telnet?
2. What principle is assigned to the J2EE security role *telnet_login*?
3. **Optional:** Assign the J2EE security role *telnet_login* to the group **GROUP-##** and test the change.

Result

For any given J2EE security role, you can trace its assignment to principles and adapt this if necessary.

Task 2: UME Roles

Create and assign UME roles.

1. Can the user **JAVA-##** launch the UME Administration Console and make changes?
2. As user **ADM200-##**, create a UME role **ManageUsers-##** which permits edit access to all users. Assign the group *GROUP-##* to this role.
3. Can the user **JAVA-##** now make changes in the UME Administration Console?

Result

You can administer UME roles and assign actions.

Solution 8: The Java Authorization Concept

Task 1: J2EE Security Roles

Evaluate the assignment of J2EE security roles

1. Can user **JAVA-##** log onto your AS Java via Telnet?
 - a) Open a command prompt (e.g. at your SAP system's operating system level).
 - b) Issue the command `telnet <hostname> <telnet_port>` (e.g. `telnet twdf1234.wdf.sap.corp 51008` for a QAS group on the host twdf1234).
 - c) Log on as user **JAVA-##**.

The should see an error message informing you that you do not have the necessary authorizations.
2. What principle is assigned to the J2EE security role *telnet_login*?
 - a) If you have not already done so, log on as the user **ADM200-##** at your system's Visual Administrator.
 - b) Navigate to the entry *Server <number> → Services → Security Provider → Runtime → Policy Configurations*.
 - c) Under *Components*, select the service *service.telnet*.
 - d) Select the J2EE security role *telnet_login*.

The J2EE security role *telnet_login* acts as a reference to the reference J2EE security role *administrators* which is assigned to the UME group *SAP_J2EE_ADMIN*. Since the user *JAVA-##* is not assigned to this group, he may not use Telnet.

Continued on next page

3. **Optional:** Assign the J2EE security role *telnet_login* to the group **GROUP-##** and test the change.
 - a) In the Visual Administrator, select the security role *telnet_login*.
 - b) Switch to change mode.
 - c) Under *Role Type*, switch to a *Security Role* and confirm the pop-up.
 - d) Under *Mappings | Group*, choose the *Add* button.
 - e) Search for a group **GROUP-##** which has already been created earlier and select this group.
 - f) Now repeat the first step of the exercise (Telnet logon) – the user **JAVA-##** can now use Telnet.

Result

For any given J2EE security role, you can trace its assignment to principles and adapt this if necessary.

Task 2: UME Roles

Create and assign UME roles.

1. Can the user **JAVA-##** launch the UME Administration Console and make changes?
 - a) Close any Web browser windows.
 - b) Enter the URL **`http://<hostname>.wdf.sap.corp:5<instance>00/useradmin`** (example for a QAS group on the host twdf1234: **`http://twdf1234.wdf.sap.corp:51000/useradmin`**).
 - c) Enter the logon data for the user **JAVA-##**.



Note: Thanks to the UME role *SAP_JAVA_NWADMIN_LOCAL_READONLY* that was previously assigned to the group *GROUP-##*, the user possesses the action *UME.Read_All* which permits read access.

Continued on next page

2. As user **ADM200-##**, create a UME role **ManageUsers-##** which permits edit access to all users. Assign the group *GROUP-##* to this role.
 - a) Log on at the UME Administration Console as user **ADM200-##**.
 - b) In the Administration Console's *Identity Management* area, select the *Roles* view.
 - c) Press *Create Role*.
 - d) In the *General Information* tab, enter **ManageUsers-##** under *Unique Name*.
 - e) Go to the *Assigned Actions* tab. Under *Available Actions* search for the action *UME.Manage_Users*. Select this entry and click *Add*.
 - f) Go to the *Assigned Groups* tab. Under *Available Groups* search for the group *GROUP-##*. Select this entry and click *Add*.
 - g) *Save* the new role.
3. Can the user **JAVA-##** now make changes in the UME Administration Console?
 - a) Log on at the UME Administration Console as user **JAVA-##** and test the possibilities.

The user **JAVA-##** can use the Administration Console and administer all the users but is not authorized to modify roles and groups (and can therefore also not assign these principles to users).

Result

You can administer UME roles and assign actions.



Lesson Summary

You should now be able to:

- Explain the terms UME role and J2EE security role
- List the authorization administration tools
- Assign actions to a UME role
- Explain how to assign J2EE security roles to users/groups

Related Information

- Online documentation for SAP NetWeaver 7.0, path: *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security Identity Management* → *Identity Management of the Application Server Java*

Lesson: Special Principles

Lesson Overview

You require special users to administer an AS Java. You can only log on to the administration tools with these users. If you have forgotten the password of the administration user, or locked this user you can activate an emergency user that can still log on.



Lesson Objectives

After completing this lesson, you will be able to:

- List a number of “special” principles
- Change the password of the administration user
- Activate the emergency user

Business Example

You are using Java applications that run on AS Java. The (only) administration user has been locked due to failed logon attempts and no further administrative activities can be performed. In this case, you need to activate the emergency user.

Default Principles

During AS Java installation, certain principles are created for special purposes while others are created subsequently by the administrator. In this section you will get to know some of these “default principles”. In some cases, the default IDs of these principles depend on the configured data source.

Default Users

The following table presents important default users:



Default Users

User	Data Source			
	Database	LDAP Server	ABAP System	
			Add-In (ABAP+Java)	Remote
Administration user	<i>Administrator</i>	<i>Administrator</i>	<i>J2EE_ADMIN</i>	<i>J2EE_ADMIN_<SID></i>
Guest user	<i>Guest</i>	<i>Guest</i>	<i>J2EE_GUEST</i>	<i>J2EE_GST_<SID></i>
Communication user to data source	<i>SAP<SID>DB</i>	Freely definable	<i>SAPJSF</i>	<i>SAPJSF_<SID></i>

The administration user has unrestricted access to AS Java and you should therefore assign this account to only very few people and assign a carefully chosen password.

If you use a client of an ABAP system as the data source, the listed user master records are located on this ABAP client (and can be viewed in SU01): In the case of a remote ABAP system, the SID of the AS Java system is incorporated in the user name. This allows you to distinguish between users if multiple AS Java systems are connected to a single ABAP client.

Among other things, the guest user is used for anonymous access to AS Java, for example in order to construct the logon form in the Web browser. This user is normally locked. Do not delete this user.

Default Groups

The following table presents important default groups:



Default Groups

Group	Data Source		
	Database	LDAP Server	ABAP System
Administrators	<i>Administrators</i>	<i>Administrators</i>	<i>SAP_J2EE_ADMIN</i>
Guests	<i>Guests</i>	<i>Guests</i>	<i>SAP_J2EE_GUEST</i>

Group	Data Source		
	Database	LDAP Server	ABAP System
All Users	<i>Everyone</i>	<i>Everyone</i>	<i>Everyone</i>
Authenticated Users	<i>Authenticated Users</i>	<i>Authenticated Users</i>	<i>Authenticated Users</i>
Anonymous Users	<i>Anonymous Users</i>	<i>Anonymous Users</i>	<i>Anonymous Users</i>

All the users you assign to the Administrator group are given extensive system authorizations (in respect both of the administrator roles assigned to this group (see next section) and the J2EE security roles associated with this group (see previous lesson)). Initially, the default administration user is entered here.

Initially, the default guest user is assigned to the guest group.

In addition, the UME possesses an integrated group adapter which is responsible for the following three special groups:

- **Everyone:** Every (!) user is always a member of this group. If you assign roles/actions to this group then every user (including those that are created in the future) has the corresponding authorizations.
- **Authenticated Users:** You assign all the users who - in whatever way - have to log onto AS Java to this group.
- **Anonymous Users:** All the users who are able to log on anonymously are assigned to this group (configured by means of the UME property *ume.login.guest_user.uniqueids*).

The following therefore applies: *Authenticated Users* + *Anonymous Users* = *Everyone*.

Default Roles

The following table presents important default roles:



Default Roles

Role	Meaning
<i>Administrator</i>	Provides extensive Java authorizations for administrators (via actions)
<i>Everyone</i>	Is shipped as an empty role and can be used by the customer to assign certain authorizations to a large number of users

Although by default no users are (directly) assigned to these two roles, the *Administrator* role is linked to the *Administrators* group. When shipped, the *Everyone* role contains no actions. You can assign authorizations (in the form of actions) to this role and then make these available to all users, for example via the *Everyone* group.

Administration User Password

You define the password for the administration user when installing an AS Java. After the installation, you can, of course, create other users with the same authorizations. However, the *one and only* administration user is special because this is not only used by the administrator in person but is also used for deployment via the SDM server:

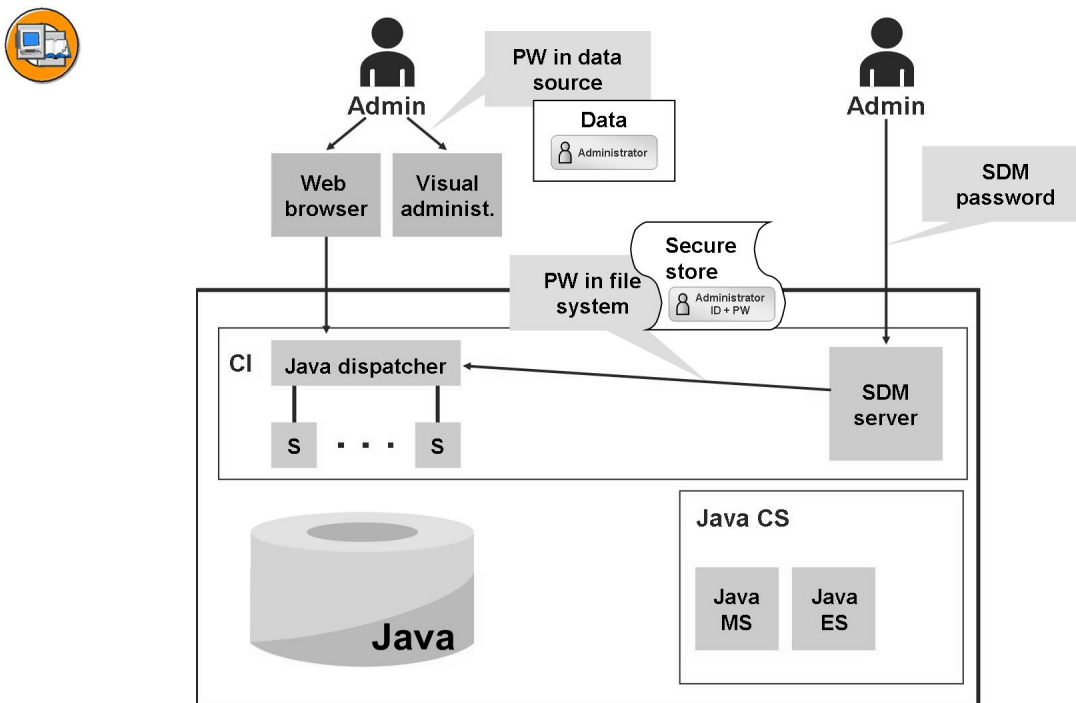


Figure 78: Special Administration User

If the SDM server performs deployment (e.g. when customer developments are imported via the NWDI or corrections with the JSPM) then it requires an administration user. It is only possible to log onto the SDM server via the “SDM password”. In addition, the SDM server cannot read any password information from the system database's Java schema.

How does the SDM server obtain the required administration authorizations? To do this, the SDM server accesses **secure storage** which is implemented as a file in the file system. This contains – among other things – the user and password of the *single* administration user.



Caution: Changes to the administration user password must be made both in the database and in the secure storage.

The following figure shows where you have to make any changes to the administration user:

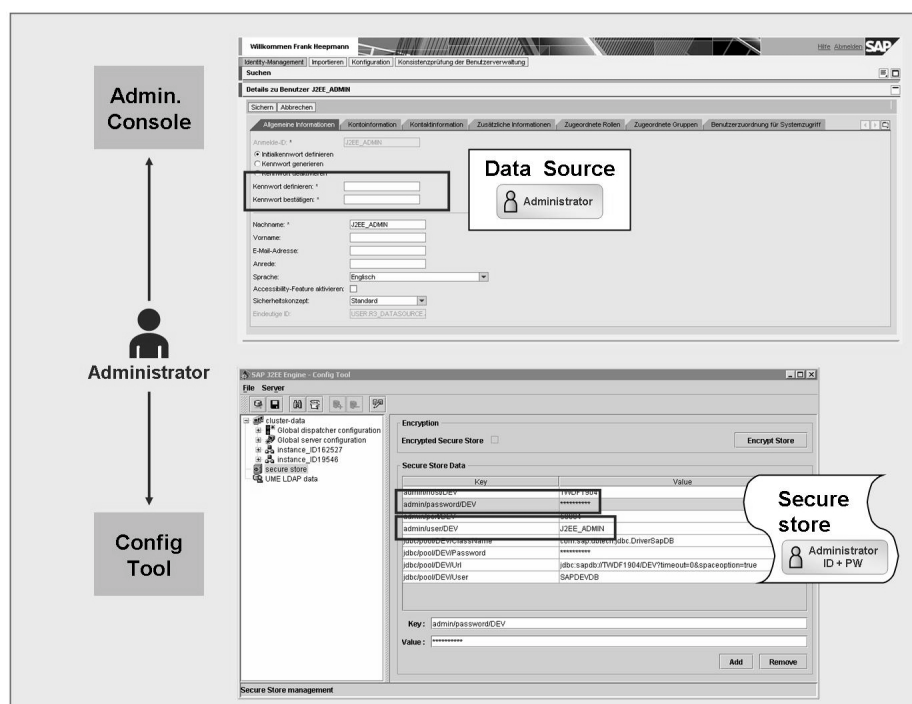


Figure 79: Making Changes to the Administration User

- You make any changes to the user master record by means of the UME Administration Console (or in the Visual Administrator or – in the case of an ABAP data source – in transaction SU01).
- You maintain the secure storage in the Config Tool (menu item *secure store*).

Emergency User

You need to activate an emergency user for the UME if the user management has been incorrectly configured and no one can log on to an application, or all administration users are locked. This emergency user is called *SAP** and can log on to any application and to the configuration tools. The *SAP** user has full administration authorizations and, for security reasons, does not have a default password. You set the password as part of emergency user activation.



Hint: The emergency user is generally not important in systems in which the UME runs (successfully) with the ABAP data source as you can always create a user in ABAP and give it Java administration rights.

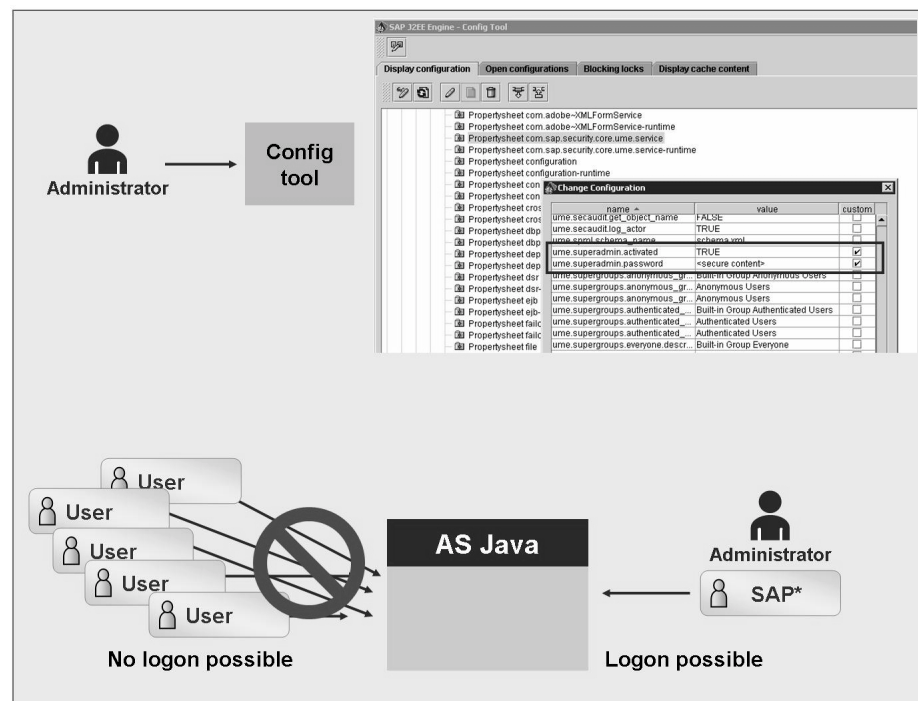



Figure 80: Activating the Emergency User

Proceed as follows to make a correction with the *SAP** user:

1. Activate the *SAP** user
 - a) Stop the Java cluster.
 - b) In the Config Tool, open the Configuration Editor mode.
 - c) Navigate to *cluster_data* → *Server* → *cfg* → *services* → *Property sheet* *com.sap.security.core.ume.service*
 - d) Switch to change mode.
 - e) Set *ume.superadmin.activated* to the value **true**.
Set *ume.superadmin.password* to any password.
 - f) Start the Java cluster.
2. Change the configuration
 - a) Log on with the user **SAP*** and the password that you have just set
 **Note:** While the *SAP** user is active, all other users are deactivated
 - b) Correct the problem; for example, unlock the administration user
3. Deactivate the *SAP** user
 - a) Stop the Java cluster.
 - b) In the Config Tool, open the Configuration Editor mode.
 - c) Navigate to *cluster_data* → *Server* → *cfg* → *services* → *Property sheet* *com.sap.security.core.ume.service*
 - d) Switch to change mode.
 - e) Set *ume.superadmin.activated* to the value **false**.
 - f) Start the Java cluster.

Exercise 9: Default Principles and Emergency Users

Exercise Objectives

After completing this exercise, you will be able to:

- Evaluate default principles
- Activate the emergency user

Business Example

You are using a Java application that runs on AS Java. You have locked the most important administration user with failed logon attempts, and now cannot perform any administrative activities. In this case, you need to activate the emergency user.

Task 1: Default Groups

Evaluation of the groups assigned to a user.

1. What UME groups are assigned to the user **JAVA-##**? Which of these are default groups?

Result

You can evaluate the default groups which are assigned to a user.

Task 2: Emergency User

Activate (and deactivate) the UME emergency user.

1. Stop all the nodes in your Java cluster.



Note: You do not have to stop the Central Services instance.

2. Activate the UME emergency user.
3. Start all the nodes of the central instance of your Java cluster.
4. Attempt to log on with user **ADM200-##** at the UME Administration Console and the Visual Administrator.
5. Attempt to log on with user **SAP*** at the UME Administration Console and the Visual Administrator.
6. Deactivate the UME emergency user.

Continued on next page

Result

You can activate the UME emergency user.

Solution 9: Default Principles and Emergency Users

Task 1: Default Groups

Evaluation of the groups assigned to a user.

1. What UME groups are assigned to the user **JAVA-##**? Which of these are default groups?
 - a) Enter the URL **http://<hostname>.wdf.sap.corp:5<instance>00/useradmin** (example for a QAS group on the host twdf1234: **http://twdf1234.wdf.sap.corp:51000/useradmin**).
 - b) Enter the logon data for the user **ADM200-##**.
 - c) In the Administration Console's *Identity Management* area, run a search for the user **JAVA-##**.
 - d) Select the hit **JAVA-##**.
 - e) Go to the *Assigned Groups* tab.

If you perform a search with the *Search Recursively* field selected, all the assigned groups will be listed. In the *Integrated Group Adapter* above the *Search Criteria*, you will see the default groups *Everyone* and *Authenticated Users* to which this user is assigned.

Result

You can evaluate the default groups which are assigned to a user.

Task 2: Emergency User

Activate (and deactivate) the UME emergency user.

1. Stop all the nodes in your Java cluster.



Note: You do not have to stop the Central Services instance.

- a) Log on at the AS ABAP via the SAP GUI with user **ADM200-##**.
- b) Start transaction SMICM.
- c) Select *Administration* → *J2EE Cluster (global)* → *Send Soft Shutdown* → *Without Restart*.

Continued on next page

2. Activate the UME emergency user.
 - a) Start the Config Tool at your SAP systems operating system.
 - b) Go to *Switch to configuration editor mode*.
 - c) Navigate to *cluster_data → Server → cfg → services → Property sheet com.sap.security.core.ume.service*
 - d) Set the parameter *ume.superadmin.activated* to the value **true** and the parameter *ume.superadmin.password* to any password.
3. Start all the nodes of the central instance of your Java cluster.
 - a) Identify the instance with which your SAP GUI is connected (*System → Status → Host Data → Server Name*). If you are not logged on to the central instance, use SM51 to switch to the central instance.
 - b) In transaction SMICM, select the path *Administration → J2EE Cluster (local) → Restart → Yes*.
4. Attempt to log on with user **ADM200-##** at the UME Administration Console and the Visual Administrator.
 - a) Both attempts fail with the message “User SAP* is active”.
5. Attempt to log on with user **SAP*** at the UME Administration Console and the Visual Administrator.
 - a) Both calls succeed. In the UME Administration Console, the user *SAP** can call all the principles. For the Visual Administrator, you need to create a new connection entry.
6. Deactivate the UME emergency user.
 - a) Stop all the nodes in your Java cluster that are still running (transaction SMICM).
 - b) Use the Config Tool to reset the parameter *ume.superadmin.activated* to its shipped value **false** (*Restore default* button).
 - c) Start all the nodes in your Java cluster (transaction SMICM).

Result

You can activate the UME emergency user.



Lesson Summary

You should now be able to:

- List a number of “special” principles
- Change the password of the administration user
- Activate the emergency user

Related Information

- Online documentation for SAP NetWeaver 7.0, path *SAP NetWeaver Library* → *Administrator's Guide* → *SAP NetWeaver Security Guide* → *Security Guides for SAP NetWeaver According to Usage Types* → *Security Guide for Usage Type AS* → *SAP NetWeaver Application Server Java Security Guide* → *User Administration and Authentication* → *User Administration and Standard Users*
- Online documentation for SAP NetWeaver 7.0, path *SAP NetWeaver Library* → *SAP NetWeaver by Key Capability* → *Security Identity Management* → *Identity Management of the Application Server Java* → *Troubleshooting* → *Activating the Security User*



Unit Summary

You should now be able to:

- List the various UME data sources
- Determine the current data source assignment
- Explain the term UME data partitioning
- Identify and modify configuration parameters
- List and use the tools for administering users and groups
- Explain the terms UME role and J2EE security role
- List the authorization administration tools
- Assign actions to a UME role
- Explain how to assign J2EE security roles to users/groups
- List a number of “special” principles
- Change the password of the administration user
- Activate the emergency user



Test Your Knowledge

1. Which of the following data sources are supported by the UME:

Choose the correct answer(s).

- ☐ A Database
- ☐ B Filesystem
- ☐ C ABAP User Management
- ☐ D Directory service

2. What is the purpose of the data partitioning of the UME?

3. You can lock users with the UME console.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

4. You can assign permissions directly to users in the UME Administration Console.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

5. The term J2EE security role is another name for a UME role.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

6. If the emergency user (*SAP**) is activated, the administration user (*Administrator*, *J2EE_ADMIN* or *J2EE_ADMIN_<SID>*) can also log onto AS Java.

Determine whether this statement is true or false.

- ☐ True
- ☐ False



Answers

1. Which of the following data sources are supported by the UME:

Answer: A, C, D

These three types of data source are available for the UME.

2. What is the purpose of the data partitioning of the UME?

Answer: The data partitioning allows a distribution of the users or user attributes to different data sources.

3. You can lock users with the UME console.

Answer: True

The UME console allows you to administer users.

4. You can assign permissions directly to users in the UME Administration Console.

Answer: False

Permissions are combined into actions, and the administrator then combines these into roles. Roles can be assigned to a user.

5. The term J2EE security role is another name for a UME role.

Answer: False

A J2EE security role is part of the J2EE standard and is used for a declarative authorization check. A UME role is an (SAP) extension to the J2EE standard and is used for a programmable authorization check.

6. If the emergency user (*SAP**) is activated, the administration user (*Administrator*, *J2EE_ADMIN* or *J2EE_ADMIN_<SID>*) can also log onto AS Java.

Answer: False

If the emergency user *SAP** is activated then no other users can log onto AS Java.

Unit 4

RFC Connections

Unit Overview

In this unit, you will learn about remote connections, also known as Remote Function Calls (RFC). As well as the various options for using the RFC, you will learn about the technical setup for connections of this type.



Unit Objectives

After completing this unit, you will be able to:

- Explain the principle of the Remote Function Call
- List the different types of Remote Function Call
- Set up an RFC connection
- Monitor RFC connections

Unit Contents

Lesson: Fundamentals and Variants for Using RFC	232
Lesson: Setting Up RFC Connections	237
Exercise 10: Set up Remote Connections	241

Lesson: Fundamentals and Variants for Using RFC

Lesson Overview

This lesson will provide you with an overview of Remote Function Calls.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the principle of the Remote Function Call
- List the different types of Remote Function Call

Business Example

SAP systems can communicate with each other using Remote Function Calls. A prerequisite for this is that the administrator has set up the relevant interface system.

RFC Fundamentals

Remote Function Calls have been used for many years as the technical interface with which SAP and non-SAP systems are usually connected. It is irrelevant whether data exchange is synchronous or asynchronous, periodic or aperiodic, or transactional. Many conceivable variants are supported.

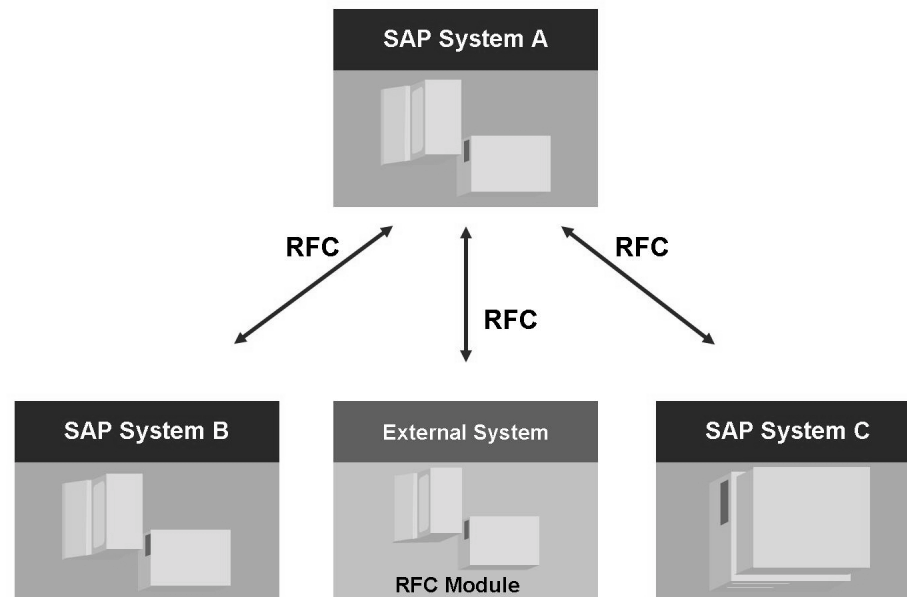


Figure 81: The RFC Interface

A “Remote Function Call” (RFC) is the call of a function module that is running in a different system to the calling program. You can call a function module in the same system as an RFC too. However, RFCs are normally used when the calling and called function modules are running in different systems.

In the SAP system, the RFC interface system provides this function. The RFC interface system allows function calls between two SAP systems or between an SAP system and an external (non-SAP) system.

RFC is an SAP interface protocol that is based on the Common Programming Interface for Communication (CPI-C) and allows cross-host communication between programs. This enables external applications to call ABAP functions and SAP systems to contact (RFC-enabled) external applications.

RFC means that ABAP programmers do not have to write their own communication routines. For an RFC call, the RFC interface

- converts all parameter data to the format required in the remote system
- calls the communication routines that are required to communicate with the remote system
- handles errors that occur during the communication

The RFC interface is easy for the ABAP programmer to use. The processing steps for calling external programs are integrated into the CALL FUNCTION statement.

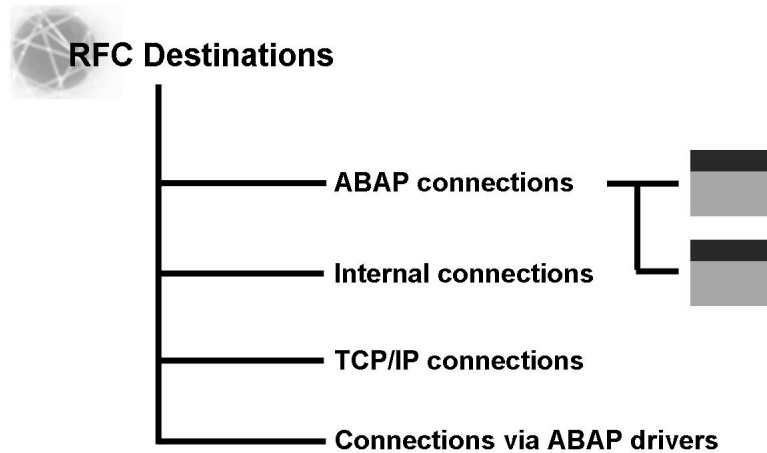


Figure 82: RFC Connections

To be able to call a function module on a remote system, you must define the remote system as a destination in your calling system. You also require access authorization for the remote system.

You can manage these remote connections in the calling system. To do this, call the *Display and Maintain RFC Destinations* screen, either by choosing the menu path *Tools → Administration → Administration → Network → RFC Destinations* or by calling transaction SM59 directly. The connection types and all existing destinations are displayed in a tree structure on the initial screen. For details about all available connection types, see the documentation.

There is a search function for destinations that have already been set up. To search for a destination, choose *Search* and enter your selection. The system displays a list of all matching entries. You can display all available information for each entry.

To change an existing RFC destination, select the relevant RFC destination in the menu tree and then choose *Change*.



Hint: To copy an existing RFC connection you first have to call the change screen for the RFC connection you want to copy. Then choose *Connection → Copy*.

Outlook: RFC Usage Variants



Synchronous RFC (sRFC)

For communication between different systems and between SAP Web AS and SAP GUI

Asynchronous RFC (aRFC)

For communication between different systems and for parallel processing of selected tasks.

Transactional RFC (tRFC)

A special form of asynchronous RFC. Transactional RFC ensures “transaction-like” processing of processing steps that were originally autonomous.

queue(d) RFC (qRFC)

Queued RFC is an extension of tRFC. It also ensures that individual steps are processed in sequence.

“RFC” is a superordinate term for various implementation variants. **sRFC** is the synchronous call of function modules. This means that the client waits until the server has completed its processing. Within an SAP system, an RFC can also be executed asynchronously in another work process. This variant is called **aRFC**.

There is also **tRFC**, the transactional Remote Function Call. Transactional RFC is asynchronous and ensures that data that is sent more than once due to network problems can be recognized at the server side, by assigning a Transaction Identifier (TID). This allows you to prevent data being processed more than once, leading to erroneous information in the application. Due to the asynchronous processing, however, parameters can only be transferred from the client to the server in this case. Returning information or status information directly is not possible.

qRFC with Send Queue is an extension of tRFC. It creates a layer between applications and the tRFC and only allows the tRFC to transfer a Logical Unit of Work (LUW) to the target server when its predecessors are no longer in the associated wait queues. After a qRFC LUW is executed, the qRFC manager automatically processes the next waiting qRFC LUW in accordance with the sequence in the wait queue.



Lesson Summary

You should now be able to:

- Explain the principle of the Remote Function Call
- List the different types of Remote Function Call

Lesson: Setting Up RFC Connections

Lesson Overview

In this lesson, you will learn how to set up a remote connection.



Lesson Objectives

After completing this lesson, you will be able to:

- Set up an RFC connection
- Monitor RFC connections

Business Example

As part of an e-commerce scenario, functions from different SAP systems must be linked with each other. Posting data, for example, is to be further processed in another system.

Remote Connections

To create a new RFC destination, choose the *Create* pushbutton in transaction SM59 (*Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations*). The system displays a new screen with empty fields that you must fill out.



The screenshot shows the 'RFC Destination' configuration dialog box. At the top, 'RFC Destination' is set to 'RFC-QAS' and 'Connection Type' is set to '3' (R/3 connection). Below this is a 'Description' field containing 'Remote connection to QAS'. The dialog has three tabs: 'Technical Settings' (selected), 'Logon/Security', and 'Special Options'. Under 'Technical Settings', 'Load Distribution' has radio buttons for 'Yes' and 'No' (selected). 'Target Host' is 'Host name QAS' and 'System Number' is '00'. 'Save As' has radio buttons for 'Host Name' and 'IP Address' (selected). At the bottom, the 'Gateway Options' section has fields for 'Gateway Host' and 'Gateway Service', and a 'Delete' button.

Figure 83: Setting Up an RFC Connection

The system opens the dialog for creating a new RFC destination.

Enter a name for the destination, for example, the connection type **3**, and a short description.

Choose *Save*. The system saves all your entries and switches to the technical settings screen. Alternatively, you can also choose **Return** here, however, your entries will not be saved if you do so.

Enter the target host and the system number of the system and choose *Save* (*Control+S*).

For a quicker logon, you can specify a client, user name, and password for logon to the target system in the *Logon/Security* tab page. Do not use your own user data here, but rather general user data, as every user (with the appropriate RFC authorizations) can use the RFC destination that you create.



Caution: Make sure you make an entry in the Client field for two reasons in particular:

1. Without specifying a target client, it may be the case that your defined RFC connection cannot be used as you expect, in spite of the connection having been tested successfully. If this is the case, it is fairly difficult to find the cause of the error, since the error messages do not point to the missing entry in the *Client* field.
2. As you can see, RFC connections between ABAP-based SAP systems always target a certain client. Thus, they do not communicate “with a particular system”, but rather “with a selected client in a particular system”.



Hint: RFC connections can always be used across the entire system. This means that an RFC connection you have defined in client 000 can also be used from client 100 (without any difference).

For security reasons, you should leave the fields *User* and *Password* empty, or enter a “Communication”-type user with very specific (in other words, adjusted to requirements) authorizations. In the first case, the system displays an input prompt for logon when you later create a connection; in the second case, dialog logons to the system are not possible, although programs can use the connection to communicate.



Note: The *PW Status* field informs you whether you have already stored a password in the masked *Password* field or not.

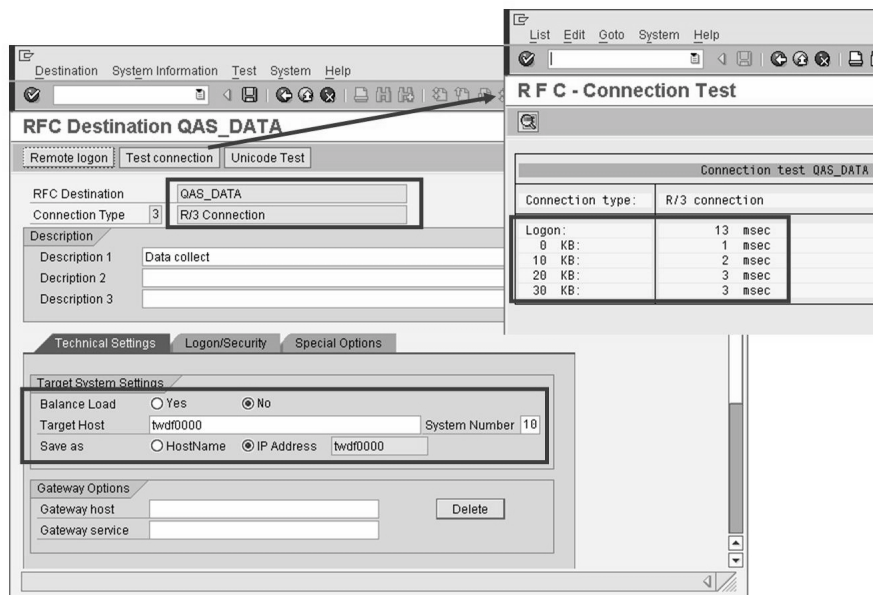


Figure 84: Testing RFC Connections

You have two options for testing a destination:

- You can attempt to log on to the remote system. To do this, choose *Remote Login*. A new session opens for the remote system. Enter the client, your user name, and your password. If you have stored a dialog user with password in the connection, a dialog logon is performed.

If you have stored a communication user, you can check that the specified password is correct with the function *Utilities* → *Test* → *Authorization Test*.

- With a connection test (*Test Connection* button or *Utilities* → *Test* → *Connection Test*), the system attempts to create a connection to the target system and displays a table with response times. If an error message appears, check your settings. This test is a pure “technical” connection test, and only checks whether a partner system can be reached with the specifications you have made.

Exercise 10: Set up Remote Connections

Exercise Objectives

After completing this exercise, you will be able to:

- Create and monitor a remote connection

Business Example

As an administrator, you are to set up remote connections to other systems, which are then to be used, for example, in the context of an ALE integrated system.

Task 1: Create the First Remote Connection

Set up a remote connection.

1. You are to create a remote connection to the central instance of the second system on your host, to **client 000**.

Use the name **DATA_000_<SID>**.

The group that is using the DEV system sets up a connection to the QAS system.

The group that is using the QAS system sets up a connection to the DEV system.

Ask your partner group for a user and password on the target system.

You (and your partner group) may have created the user **CSMREG** in a previous exercise. If this is available, use it.

Result

You have created an initial RFC connection.

Task 2: Create the Second Remote Connection

Set up another remote connection.

1. You are to create a remote connection to the central instance of the second system on your server, this time to **client 100**.

Use the name **ANALYSIS_100_<SID>**.

The group that is using the DEV system sets up a connection to the QAS system.

The group that is using the QAS system sets up a connection to the DEV system.

Leave the Client, User and Password fields empty.

Continued on next page

For a remote logon test, get a user with a password on the target system from your partner group.

Result

You have created another RFC connection.

Task 3: Test the Remote Connections

Test the remote connections you have created by choosing *Test Connection*.

Check the authorizations of the defined user of the connection **DATA_000_<SID>** by choosing *Utilities → Test → Authorization Test*.

1. Test the connections **DATA_000_<SID>** and **ANALYSIS_100_<SID>**.
2. Attempt to log on to the remote system.

Use the function *Utilities → Test → Authorization Test*.

Result

The RFC connections that you have created work correctly and can be used later for central system monitoring.

Task 4: Use Load Balancing (Optional)

Use the load balancing for an RFC connection.

1. Create an additional RFC connection to your partner group's system. Call this connection **<SID>_GROUP**. Use the option for load balancing, by logging on using a logon group, for this connection. To do this, you require the name of the logon group that your partner group set. Alternatively, you can use the default group **SPACE**.

Result

You can now successfully set up an RFC connection using load balancing.

Solution 10: Set up Remote Connections

Task 1: Create the First Remote Connection

Set up a remote connection.

1. You are to create a remote connection to the central instance of the second system on your host, to **client 000**.

Use the name **DATA_000_<SID>**.

The group that is using the DEV system sets up a connection to the QAS system.

The group that is using the QAS system sets up a connection to the DEV system.

Ask your partner group for a user and password on the target system.

Continued on next page

You (and your partner group) may have created the user **CSMREG** in a previous exercise. If this is available, use it.

- a) To set up a remote connection, switch to the window *Display and Maintain RFC Destinations*. To do this, choose *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations* (transaction SM59) and then *Create*.

The system opens the dialog for creating a new RFC destination. Enter the following:

Field	Entry
RFC Destination	DATA_000_<SID> , using the SID of your partner system
Connection Type	3
Description 1	Connection for central monitoring, to collect data

Choose *Save*. The system stores your entries and the technical settings screen appears.

Field	Input
Target Host	<Host Name> , name the server on which your partner system is running
System Number	00 or 10 , depending on the partner system SID

Choose *Save* (**CTRL+S**).

Store the following data on the *Logon/Security* tab page.

Field	Input
Client	000
User	CSMREG , created by the partner group.
Password	The password for the user CSMREG .

Save your entries.

Result

You have created an initial RFC connection.

Continued on next page

Task 2: Create the Second Remote Connection

Set up another remote connection.

1. You are to create a remote connection to the central instance of the second system on your server, this time to **client 100**.

Use the name **ANALYSIS_100_<SID>**.

The group that is using the DEV system sets up a connection to the QAS system.

The group that is using the QAS system sets up a connection to the DEV system.

Leave the Client, User and Password fields empty.

For a remote logon test, get a user with a password on the target system from your partner group.

- a) To set up a remote connection, switch to the window *Display and Maintain RFC Destinations*. To do this, choose *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations* (transaction SM59) and then *Create*.

The system opens the dialog for creating a new RFC destination. Enter the following:

Field	Input
RFC Destination	ANALYSIS_100_<SID> , using the SID of your partner system
Connection Type	3
Description 1	Connection for central monitoring, to analyze problems

Choose *Save*. The system saves your entries and the technical settings screen appears.

Field	Input
Target Host	<Host Name> , name the server on which your partner system is running
System Number	00 or 10 , depending on the partner system SID

Choose *Save* (*CTRL+S*).

Store the following data on the *Logon/Security* tab page.

Continued on next page

Field	Input
Client	<blank>, do not enter anything here
User	<blank>, do not enter anything here
Password	<blank>, do not enter anything here

Save your entries.

Result

You have created another RFC connection.

Task 3: Test the Remote Connections

Test the remote connections you have created by choosing *Test Connection*.

Check the authorizations of the defined user of the connection **DATA_000_<SID>** by choosing *Utilities* → *Test* → *Authorization Test*.

1. Test the connections **DATA_000_<SID>** and **ANALYSIS_100_<SID>**.

- a) Choose the *Test Connection* pushbutton for the connections **DATA_000_<SID>** and **ANALYSIS_100_<SID>**.

The system attempts to create a connection to the remote system, and, if the connection is correctly set up and the target system is accessible, displays a list of response times.

2. Attempt to log on to the remote system.

Continued on next page

Use the function *Utilities → Test → Authorization Test*.

- a) Choose the *Remote Logon* pushbutton for the connection **DATA_000_<SID>**.

No logon is performed for the connection **DATA_000_<SID>**, since the user that you have specified (**CSMREG**) is not a dialog-capable user in the remote client.

Use the menu to call the function *Utilities → Test → Authorization Test*.

If you have stored the correct logon data, this test is performed without an error message.

Choose the *Remote Logon* pushbutton for the connection **ANALYSIS_100_<SID>**.

No automatic logon is performed for the connection **ANALYSIS_100_<SID>**, since you have not defined any logon data. If you have valid logon data, you can enter this data and log on successfully.

Result

The RFC connections that you have created work correctly and can be used later for central system monitoring.

Continued on next page

Task 4: Use Load Balancing (Optional)

Use the load balancing for an RFC connection.

1. Create an additional RFC connection to your partner group's system. Call this connection **<SID>_GROUP**. Use the option for load balancing, by logging on using a logon group, for this connection. To do this, you require the name of the logon group that your partner group set. Alternatively, you can use the default group **SPACE**.

- a) Call the maintenance transaction for RFC connections (*Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations*, transaction SM59).

Choose *Create*. Enter the following values:

- *RFC Destination*: **<SID>_GROUP** (replace <SID> with the <SID> of your partner group's system).
- *Connection type*: **3**
- *Description*: Any documentation.
- Choose *Enter*.
- Select the radio button *Load Balancing: Yes*
- *Target host*: The host of your partner group.
- Choose *Enter*.
- *Group*: Logon group that your partner group set up or the default group **SPACE**.

On the *Logon/Security* tab page, enter the logon information that you received from your partner group.

Choose *Save*.

Result

You can now successfully set up an RFC connection using load balancing.



Lesson Summary

You should now be able to:

- Set up an RFC connection
- Monitor RFC connections



Unit Summary

You should now be able to:

- Explain the principle of the Remote Function Call
- List the different types of Remote Function Call
- Set up an RFC connection
- Monitor RFC connections



Test Your Knowledge

1. Which Remote Function Call procedures does an SAP system provide?

Choose the correct answer(s).

- ☐ A Synchronous RFC
- ☐ B Reflexive RFC
- ☐ C Looped RFC
- ☐ D Transactional RFC
- ☐ E Direct RFC
- ☐ F Queued RFC

2. Which RFC variant can you use to process work steps in parallel?

3. To connect two SAP systems by RFC, you require an _____
in each system (this automatically exists) and an explicitly defined
_____ from one system to the other.

Fill in the blanks to complete the sentence.



Answers

1. Which Remote Function Call procedures does an SAP system provide?

Answer: A, D, F

In addition to the three listed in the exercise (synchronous, transactional, and queued RFC), there is also asynchronous RFC.

2. Which RFC variant can you use to process work steps in parallel?

Answer: You can use asynchronous RFC to process program steps in parallel, as long as there are work processes available in the system.

3. To connect two SAP systems by RFC, you require an RFC interface in each system (this automatically exists) and an explicitly defined RFC connection from one system to the other.

Answer: RFC interface, RFC connection

The basic requirement is the RFC interface, which is in the protocol stack of every SAP system. You must also set up a connection from the calling system to the called system (transaction SM59).

Unit 5

Communication and Integration Technologies

Unit Overview

There is a vast array of methods for connecting SAP systems with other systems, and optimizing processes within a system. Many of the available technologies are briefly introduced in this unit, and you should get an idea of the various uses for each. The previous unit introduced, in great detail, an important communication technology known as the RFC. For the sake of completeness, this unit also contains information about the RFC.



Unit Objectives

After completing this unit, you will be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept
- List various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Make a Remote Function Call
- Explain the evolution from *SAP R/3* to *SAP ERP* and the Enterprise SOA
- Describe the significance of the Web services within the Enterprise SOA
- Explain Web services
- Describe UDDI and WSDL
- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept

Unit Contents

Lesson: Cross-System Business Processes	255
Lesson: Remote Function Calls and BAPIs	260
Exercise 11: Remote Function Calls and BAPIs	267
Lesson: Enterprise Services-Oriented Architecture (Enterprise SOA).....	270
Lesson: Web Services	276
Lesson: SAP Business Workflow	280
Exercise 12: Leave Request as Workflow	285

Lesson: Cross-System Business Processes

Lesson Overview

This lesson explains the fundamentals of the Application Link Enabling (ALE) concept.



Lesson Objectives

After completing this lesson, you will be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept

Business Example

Your company wants to implement an Internet sales scenario in the context of the *SAP CRM* solution.

The Significance of Cross-System Business Processes

Let's start by defining cross-system business processes, using common situations as examples.

For example, it may be the case that within a company, the human resources system is separate from the rest of the business software system. Obviously, the systems cannot be completely separate, since the accounting system needs the employees' wage data. In this situation, you need cross-system business processes to exchange the relevant data.

Cross-system business processes are used, for example, if two companies collaborate closely and send joint orders to a vendor. The companies' business IT systems need to communicate with each other to consolidate the quantities to be ordered. In this case, the business process does not just cross system boundaries, but also company boundaries.

An additional example is the transfer of a limited quantity of specific data, for example, the electronic transfer of account statement data from a bank to a company.

Recent developments suggest that the significance of cross-system business processes will continue to increase rapidly.

Application Link Enabling (ALE)

Application Link Enabling is a means of creating and operating distributed applications. The basic concept of Application Link Enabling is to ensure operation of a distributed, yet integrated system landscape. This involves business-controlled message exchange using consistent data across loosely linked application systems. The applications are integrated through synchronous and asynchronous communication, not through a central database.

Systems that use ALE to exchange data can be located at the same company, or they may belong to different companies. One of the characteristics of ALE is that different systems are linked in business terms through secure and consistent data transfer.

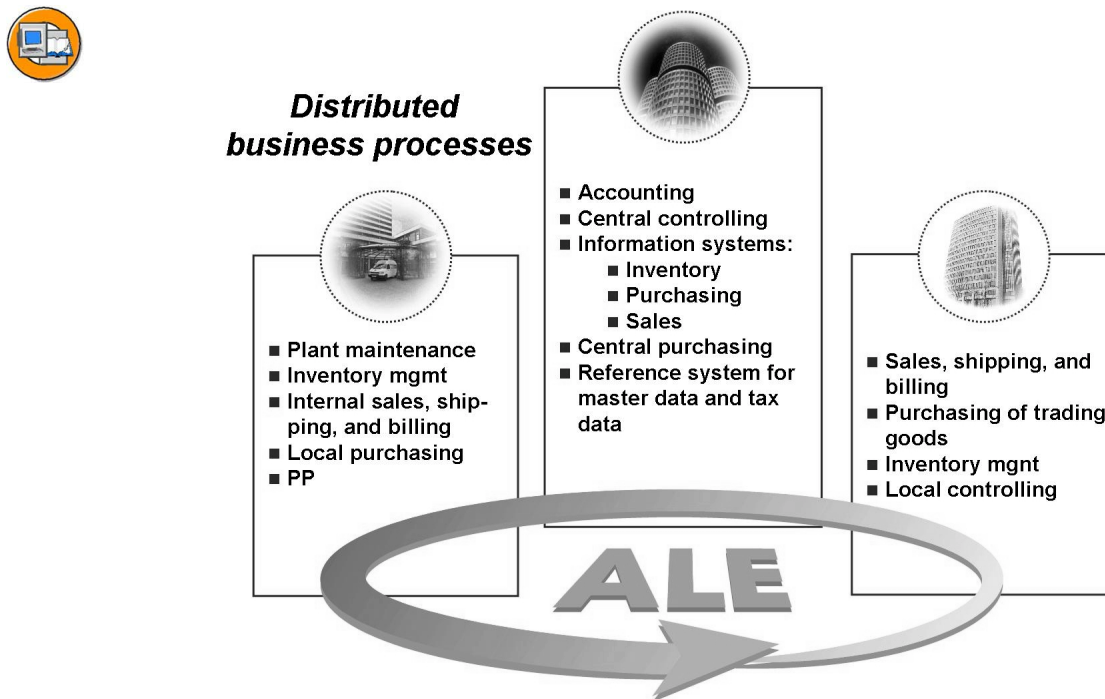


Figure 85: Business process distribution using ALE

You could also describe ALE as being composed of the elements: who exchanges which data when, with whom, and by what means.

Implementing ALE therefore requires that you clarify the following points in detail:

1. Identify the business process and the objects involved
2. Identify the information to be transmitted
3. Specify the format for the data to be transferred
4. Decide on the transfer technology to be used
5. Decide on the transfer type
6. Specify the destination of the data transfer

The following table contains examples for implementing ALE:



Data Synchronization in the Business Process – an Example

Process	Internet Sales with SAP CRM
Identify the information to be transmitted	Order data from the SAP CRM System, which is to be passed to an ERP backend
Format of the data	IDoc format
Transfer technology	by RFC
Transfer type	asynchronously, every 60 seconds
Objective	Provide goods and/or services for sale in the Internet

The data is often identified within the SAP system using a business object and its Business Application Programming Interfaces (BAPIs). A BAPI is a method of a business object, for example, the material master record. A permissible method could be creating or changing the material master data. BAPIs normally enable you to edit all data belonging to the object.

The IDoc format describes the structure of “intermediate documents”. There are various kinds of IDoc formats for different types of data to be exchanged. Alternatively, you can use ALE to transfer data in an agreed XML format.

You can select your preferred data transfer technology within the constraints imposed by the system. For example, you can transfer data by Remote Function Call (RFC) or using HTTP or HTTPS.

There are two basic types of transfer: synchronous and asynchronous. Synchronous transfer means that the data is transferred at the time of creation or change. You can start asynchronous transfers at intervals of your choice.

There are very few restrictions on the systems that can be linked. The systems involved must have the technical capability to receive the communications (RFC-enabled, HTTP-enabled) and interpret the format transferred (IDoc, XML). SAP systems of different releases can be linked using ALE.



Lesson Summary

You should now be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept

Related Information

- **BIT300** and **BIT350** are more advanced training courses on ALE.
- You can find additional information on the topic of ALE on the *SAP Service Marketplace* using the quick link */ibf* (and then under *Ibf in Detail => Integration Scenarios*).

Lesson: Remote Function Calls and BAPIs

Lesson Overview

This lesson provides an overview of the interface technologies available to you in the SAP system, while focusing on the significance of RFCs and BAPIs.



Lesson Objectives

After completing this lesson, you will be able to:

- List various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Make a Remote Function Call

Business Example

You need to integrate existing applications with SAP applications. The interfaces available in the standard system are of particular interest here.

Overview of Interfaces

SAP systems have interfaces at different communication levels. These range from highly technical connection options, for example, using the TCP/IP protocol or CPI-C, to highly specialized interfaces designed for business objects, such as BAPIs or the IDoc interface used in the ALE environment. All higher interfaces, that is, those that access business objects or processes, use the same technology, the Remote Function Call (RFC).

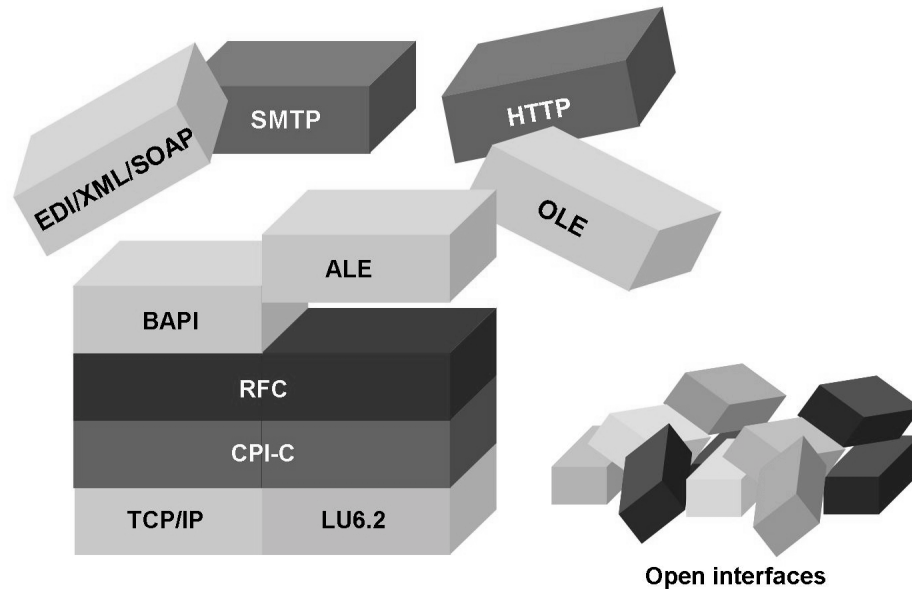


Figure 86: Interfaces technologies used in SAP systems

SAP systems use the following interface technologies that are listed in the above graphic:

- ALE: Application Link Enabling
- BAPI: Business Application Programming Interface
- CPI-C: Common Program Interface Communication
- EDI: Electronic Data Interchange
- HTTP: HyperText Transfer Protocol
- LU 6.2: Logical Unit Type 6.2
- RFC: Remote Function Call
- OLE: Object Linking and Embedding
- SMTP: Simple Mail Transfer Protocol
- SOAP: Simple Object Access Protocol
- TCP/IP: Transmission Control Protocol / Internet Protocol
- XML: Extensible Markup Language

Remote Function Call

The Remote Function Call interface is an SAP interface protocol based on CPI-C and TCP/IP. It simplifies the programming of communication processes between different systems. RFCs enable you to call and execute predefined functions **in a remote system – or within the same system**. RFCs manage the communication process, parameter transfer and error handling.

RFC describes an interface, not the programming language in which the function runs. You can also use RFCs to call functions in non-SAP systems. The procedure for RFC communication between two SAP systems is that the calling system uses an RFC definition in the system called to access a specific function.

This function is normally a remote-enabled function module. You can also, depending on the release, use RFC to call functions in SAP R/2 systems.

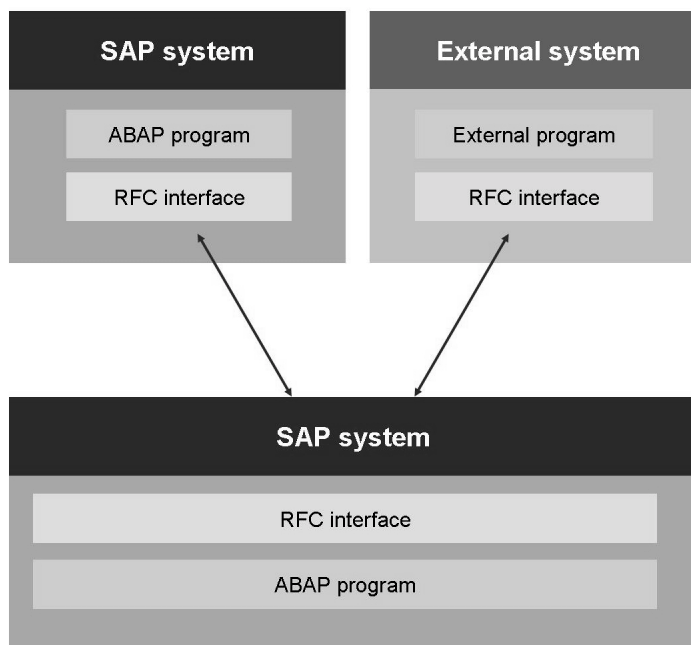


Figure 87: RFC connection possibilities

If you want to start external programs remotely, you need an RFC interface outside the SAP system. This could be, for example, a simple Dynamic Link Library (DLL). Every RFC interface is bidirectional, so external programs can also use RFC to access functions in SAP systems.



Note: All function modules (including those that are remote-enabled) are created, together with their import and export parameters, using the *Function Builder*. You can call the *Function Builder* via *Tools* → *ABAP Workbench* → *Development* → *Function Builder* or using transaction code SE37.

To call an RFC module from an SAP system, you need to know the import and export parameters (defined in the *Function Builder*), and there must be a technical connection between the two systems. This connection is called an **RFC connection** or an **RFC destination**.

You can manage your RFC connections via *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations* or using transaction SM59.

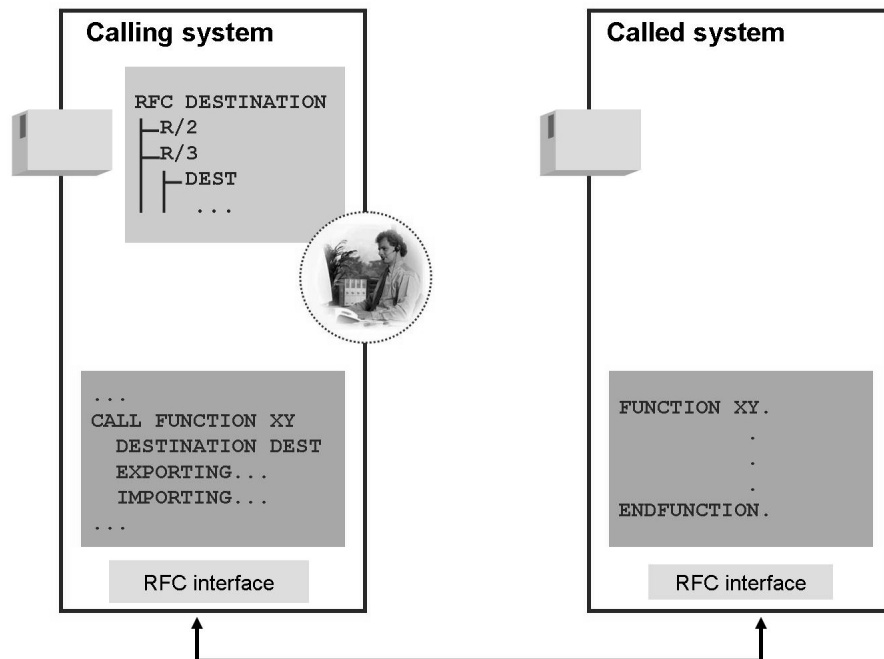


Figure 88: Remote Function Call in detail

In the above graphic you can see, on the left side, the calling system, in which an RFC destination named **DEST** has been created. An RFC destination in transaction SM59 should not be confused with an SAP system, since an RFC connection can

only point to one specific client in an SAP system. These are therefore also referred to as connections between **logical systems**; this term is used, above all, in the ALE environment.

This also means that you can have at least as many RFC connections between two systems as there are clients in the target system. Since you can specify a logon user for the destination in each RFC connection, you can therefore also access clients in the target system several times, for example, with a different logon user each time. If you need a bidirectional RFC connection between two systems, that is, that the system called can also execute RFC modules in the calling system, then you need to set up an equivalent second RFC connection in the system called.



Hint: When you are defining RFC destinations, RFC connections are

- Addressed to **one** client, when they are pointing at an SAP system
- Accessible from all clients in the system

In ABAP, you use RFCs to call a function module in another system as follows:

```
CALL FUNCTION <Name>
  DESTINATION <Ziel>
  EXPORTING ...
  IMPORTING ...
```

The function to be executed in the target system is named. The name of the target must refer to one of the RFC connections available. When you are creating an RFC connection, you can specify logon data for the target system; if you do not do this, you need to enter logon parameters when you start the RFC. `Exporting` and `Importing` are used to pass parameters to the target function and to receive the returned parameters. The function called in the target system is executed using the user ID entered for the connection.



Note: You can also create RFC connections for which the user of the user making the call is used in the target system. That means different users can use the same connection in the target system. This procedure is also called **Trusted RFC**. It is, of course, a prerequisite that identical users are created in the source and target systems. Trusted RFC is explained in the course ADM960 - *Security in SAP System Environments*.

The RFC has become the most important interface in the SAP environment. Some special RFC modules, which follow certain conventions, are also known as BAPIs (Business Application Programming Interfaces).

BOR and BAPIs

A Business Application Programming Interface (BAPI) is a standardized programming interface that facilitates internal and external access to business processes and data in SAP systems. BAPIs are defined in the Business Object Repository as methods of SAP business objects and enable an object-oriented view of business data in an SAP system. Functions that can be called using BAPIs are normally implemented and stored in the *ABAP Workbench's Function Builder* as RFC-enabled function modules. You can display an overview of available BAPIs in the BOR, for example, by activating the Business Object Repository pushbutton in the Business Object Builder (*Tools → ABAP Workbench → Development → Business Object Builder*), transaction SWO1. You can access the BOR directly using transaction code BAPI.

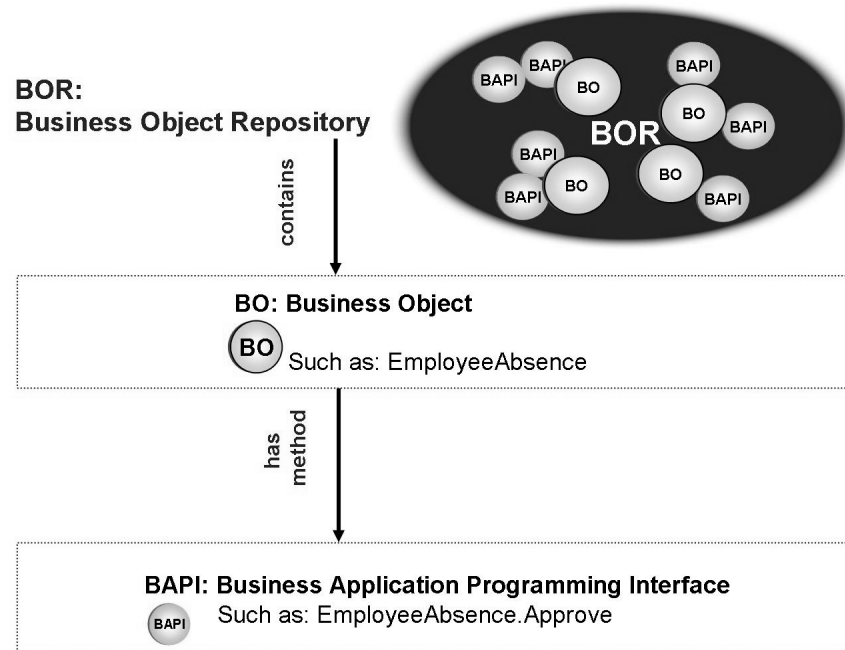


Figure 89: BOR and BAPIs

BAPIs, which represent methods for business objects in an SAP system, are used in a variety of contexts. Here are some possible uses for BAPIs:



- To link business processes across system boundaries (for example, when using ALE)
- Used by SAP to integrate various solutions in the framework of *SAP Business Suite*
- To connect an SAP system to the Internet
- Used in conjunction with SAP Business Workflow
- To connect to external programs



Note: BAPIs are created and tested in exactly the same way as other function modules, using the *Function Builder*, transaction SE37, and are then defined as BAPIs in the BOR.

Exercise 11: Remote Function Calls and BAPIs

Exercise Objectives

After completing this exercise, you will be able to:

- Use BAPIs

Business Example

You need data from another system.

Task: Using a BAPI

Use a BAPI to display the address data for your user in the system.

1. Start the overview transaction for the Business Object Repository.
2. Find the method *USER.Change*. Display the documentation for this method.
3. Call the *Function Builder* for the *USER.Display* method. You can now see the source code for the BAPI in the **Function Builder**.
4. Execute this BAPI for your user with the RFC destination **NONE**.
5. Once you have confirmed the dialog box, the result screen for the query is displayed. You can view the return code and the response time for your query.

Result

In this example, the export parameter for the calling side was the user “in the other system”, the import parameter was the receipt of the dialog box. From the point of view of the function **called**, the import parameter was the user name and the export parameter was the dialog box.

Solution 11: Remote Function Calls and BAPIs

Task: Using a BAPI

Use a BAPI to display the address data for your user in the system.

1. Start the overview transaction for the Business Object Repository.
 - a) Choose *Tools* → *ABAP Workbench* → *Development* → *Business Object Builder* and choose *Business Object Repository* (Transaction BAPI). If you are following the *Business Object Builder* menu path, then select *BAPI* in the dialog box.
2. Find the method *USER.Change*. Display the documentation for this method.
 - a) Use the *Alphabetical* tab page and look for the *User* business object.
From the BAPIs available for this business object, select the *USER.Change* BAPI. Use the appropriate tab page to display the documentation for this BAPI on the right side of the screen.
3. Call the *Function Builder* for the *USER.Display* method. You can now see the source code for the BAPI in the **Function Builder**.
 - a) Switch to the *USER.Display* BAPI. Choose the *Tools* tab page, followed by the *Function Builder*. Then choose *Display*.
4. Execute this BAPI for your user with the RFC destination **NONE**.
 - a) Press *Test/Execute* (F8).
Specify **NONE** as the RFC target system entry; this refers to your own system. Enter your user name on the query screen. Select *Execute*. As a result, your user data, requested by the RFC from the BAPI, is displayed.
5. Once you have confirmed the dialog box, the result screen for the query is displayed. You can view the return code and the response time for your query.
 - a) The return code for your query is displayed as the value for the export parameter *Return*.

Result

In this example, the export parameter for the calling side was the user “in the other system”, the import parameter was the receipt of the dialog box. From the point of view of the function **called**, the import parameter was the user name and the export parameter was the dialog box.



Lesson Summary

You should now be able to:

- List various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Make a Remote Function Call

Related Information

For information about additional interfaces, go to the following address:
<https://service.sap.com/connectors>.

Lesson: Enterprise Services-Oriented Architecture (Enterprise SOA)

Lesson Overview

This lesson aims to highlight the main differences between the software solutions *SAP R/3* and *SAP ERP* as well as the significance of the Enterprise Services-oriented Architecture (Enterprise SOA).



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the evolution from *SAP R/3* to *SAP ERP* and the Enterprise SOA
- Describe the significance of the Web services within the Enterprise SOA

Business Example

You want to understand the development of *SAP R/3* after *SAP ERP* and the Enterprise SOA, in order to be able to fully use the advantages that these offer to your company.

Enterprise Resource Planning to Date

With ERP software, that is, software for controlling and processing business-related company processes, SAP has set standards worldwide. The central idea behind the software, namely the **real-time processing** of different business processes in a company and their implementation in the successful solutions *SAP R/2* and *SAP R/3* has made SAP into a company that is active worldwide. *SAP R/3* was complemented by other solutions from SAP, such as *SAP CRM*, *SAP SCM* and *SAP SRM*.

In the past years, SAP has strongly increased the value of its own application platform and has consolidated this platform with the *SAP NetWeaver*, combining herein the technical foundations for all SAP solutions.

With *SAP NetWeaver*, all future SAP applications (including the application *SAP ERP 6.0*, for example) are provided with the basic functions, including those from other components; for example, functions of the *SAP Business Information Warehouse*.

The development of *SAP R/2* after *SAP R/3* and other solutions from the past years were characterized by different factors.

Primary differences, but also common features of SAP R/2 and SAP R/3



- In common: real-time processing of business processes
- In common: use of ABAP as a programming language, optimized for implementation in business software
- In common: constantly increasing features (during the maintenance period) by functions newly created by SAP
- In common: adjustability of mapped processes to company-specific activities
- In common: all information in a central database
- **Difference:** host-based system on client-server-based system

Thus, the change from *SAP R/2* to *SAP R/3* (and other “new” software from SAP) primarily meant a change in the technical infrastructure and in the design of the user interface. However, applications were mainly developed “in the same way”. What now?

Enterprise Resource Planning and the Enterprise SOA in the Future

With *SAP ERP 6.0*, SAP is taking the first steps towards business applications that build on an Enterprise SOA.

The tasks, which *SAP R/3* systems and other SAP software fulfil in companies worldwide, naturally also have to be fulfilled in the future. That means that the Enterprise SOA will not make any basic changes to these processes. What are the core characteristics of an Enterprise SOA?

In the usual **Client Server Architecture**, the business process data are in the system database, the application processes run on application servers and are made available via predefined interfaces. Business processes that do not belong to the classic SAP world can be integrated more easily via interfaces. The processes are very often integrated via “human integrators”. To do this, the employees of the company must know when they have to call up what systems for data maintenance in the company's business processes.

Different business applications exchange data directly via the database. For example, a financial transaction may, under certain circumstances, access an HR-related table in which data was updated from an HR transaction shortly before. In the context of Enterprise SOA, however, the financial application would request the HR data via a specific application-to-application (A2A) interface and not simply retrieve the data from the database with an SQL access.

In the **Enterprise SOA**, by comparison, there are role-based user interfaces, which act as central entry points for employees of the company to carry out their work using different applications in different systems. New process steps that are provided as **Enterprise Services** and whose data can be saved in totally different databases, can be integrated using general standards into the process world of the company with minimal effort. With the help of cross-system process definitions and process control through workflow, the amount of work for “human integrators” can be reduced.

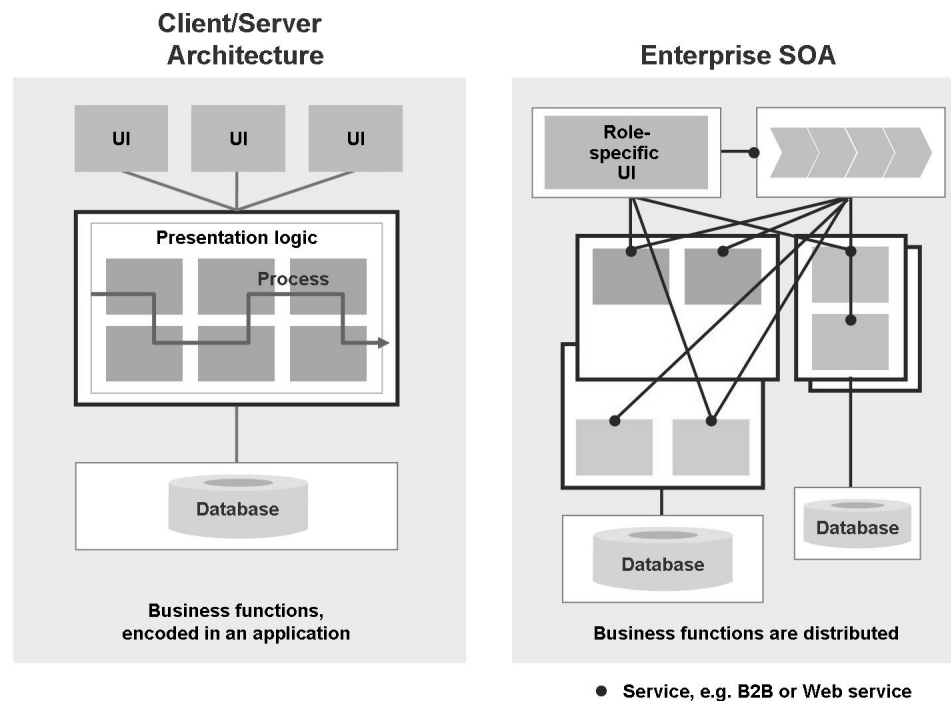


Figure 90: Client-Server Architecture Versus Enterprise SOA

The individual steps within an enterprise service can be linked via standardized, specially developed interfaces. This involves application-to-application (A2A), business-to-business (B2B) and user interface (UI) interfaces. A2A interfaces refer to links within a business application, B2B interfaces refer to links between different business applications. These designed interfaces make up the actual core of the Enterprise SOA.

All the requirements for the business application are developed top-down on the basis of component modeling, including the necessary A2A and B2B interfaces. The aim here is for there to be no functions in the system that do not originate from the model.

The interfaces resulting from the modeling can be implemented using standardized protocols. Although it is a major simplification, the technical implementations of the interfaces with standard protocols are referred to as Web services. What is a **Web Service** as opposed to an **Enterprise Service**?

Enterprise Services describe the larger business logic. An Enterprise Service does not address detail functions, but a complete, industry-specific process that can consist of many small individual steps. All actions together form the Enterprise Service, which thus provides a context-oriented business process logic. **Web Services**, by comparison, are small modular applications, which run within the framework of Internet technologies and which are generally called up as a detail function within applications or Enterprise Services. Standards were agreed to describe the call-up of Web Services. (Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP), Universal Description, Discovery, and Integration (UDDI)).

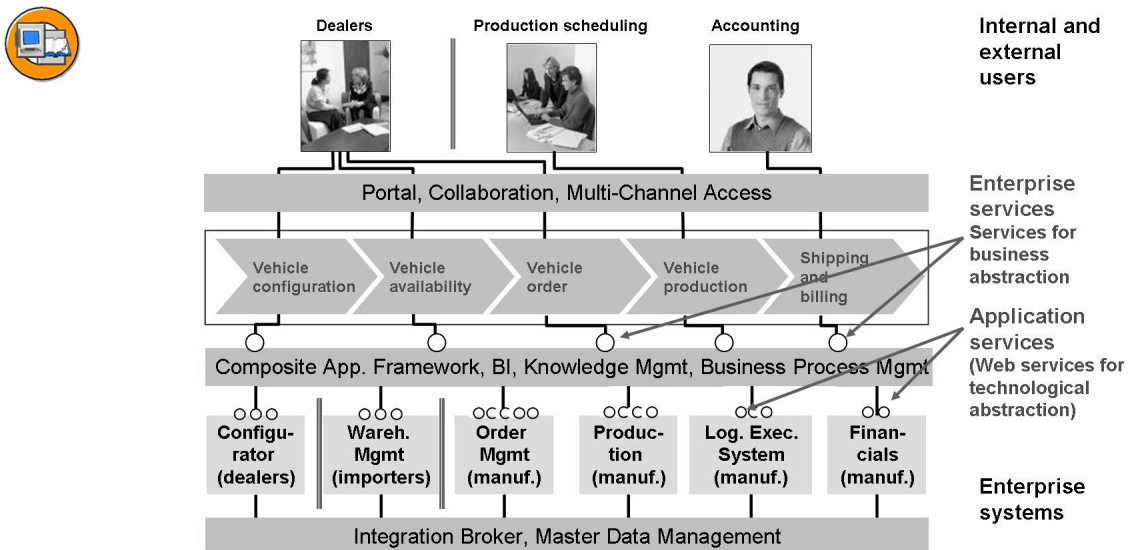


Figure 91: Enterprise SOA and Web Services



Hint: You can find more information on Web Services at <http://www.w3c.org/2002/ws>.

Enterprise SOA can be characterized using the following key ideas:

Characteristics of the Enterprise SOA (ESOA)



- An Enterprise SOA application is generally implemented across systems.
- An Enterprise SOA application is created in ABAP or in Java.
- An Enterprise SOA application generally has no “own” database.
- New functions are entered “outside” of existing systems (for Enterprise SOA applications).

The availability of Enterprise Services provides many new options.

Enterprise Services enable ...



- ... the efficient creation of new applications without having to modify the underlying system.
- ... very high flexibility in the configuration of business processes, also “in operation”.
- ... the simplified creation of applications that use the functions of several systems.

SAP will extensively offer the functions of its own software products as enterprise services and design additional new applications on the basis of these. At the same time, you will have the opportunity here to design and use new and flexible business processes across systems without having to intervene in your business systems. Of course, the Enterprise SOA also provides you with new possibilities, such as linking functions of SAP systems with the functions of enterprise or Web services from other providers via the Internet.



Lesson Summary

You should now be able to:

- Explain the evolution from *SAP R/3* to *SAP ERP* and the Enterprise SOA
- Describe the significance of the Web services within the Enterprise SOA

Related Information

- <http://service.sap.com/erp>
- <http://www.sap.com/germany/plattform/enterprisesoa/index.epx>

Lesson: Web Services

Lesson Overview

This lesson provides you with an introduction to the subject of Web services.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain Web services
- Describe UDDI and WSDL

Business Example

Your company wants to technically realize online services using Web services.

Web Services - A Short Introduction

The *SAP NetWeaver Application Server* is also a development platform for Web services. A Web service is a service that essentially provides a program interface (API). In practice, the Web service is made available via Internet protocols and the application based on it is often operated using a Web browser, although this is not the actual definition of a Web service.

Web services are the technical basis for making individual functions of an application directly available. Here, the existing function of an application can be addressed via standardized access protocols and content can also be exchanged in a structured form. In this way, cross-application functions, which are sold by SAP as complete units for the business process under the name of Packaged Composite Application (*SAP xApps*) can also be developed flexibly. The combination of several granular

services, in the sense of self-contained business scenarios, is called an enterprise service. In the *SAP NetWeaver Application Server*, the following basic standards for Web services are implemented:



- eXtensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)

SOAP describes a protocol that you can use to call up Web services in distributed system landscapes. SOAP uses HTTP as a transport protocol. An SOAP message has a header with the additional information and a body with the actual message.

- Web Service Description Language (WSDL)

WSDL is a meta language that is used to describe the function of a Web service. Functions, parameters and return codes in particular are described in a machine-readable form. WSDL is standardized by the World Wide Web Consortium (W3C); see the following URL: <http://www.w3c.org/TR/wsdl.html>

- Universal Description, Discovery, and Integration (UDDI) is a directory service for dynamic Web services. A directory of Web services is provided via an SOAP interface. The information here is highlighted in white, yellow and green pages. You can find more information on UDDI at: <http://www.uddi.org>. SAP itself also operates a UDDI server at <http://uddi.sap.com>, on which Web services can be registered and searched for.

Web Services and the SAP NetWeaver Application Server

Web services can already be developed in the SAP system from release *SAP NetWeaver Application Server 6.20*, however, the development tools have been considerably enhanced for release 6.40. Thus, an existing, remote-capable function module from the *Function Builder* can be transformed (transaction SE37) into a Web service. A Web service is a module that can be used flexibly in different applications. The creator publishes the service in a publicly accessible UDDI directory. The customer can then search directly for Web services in the UDDI directory.

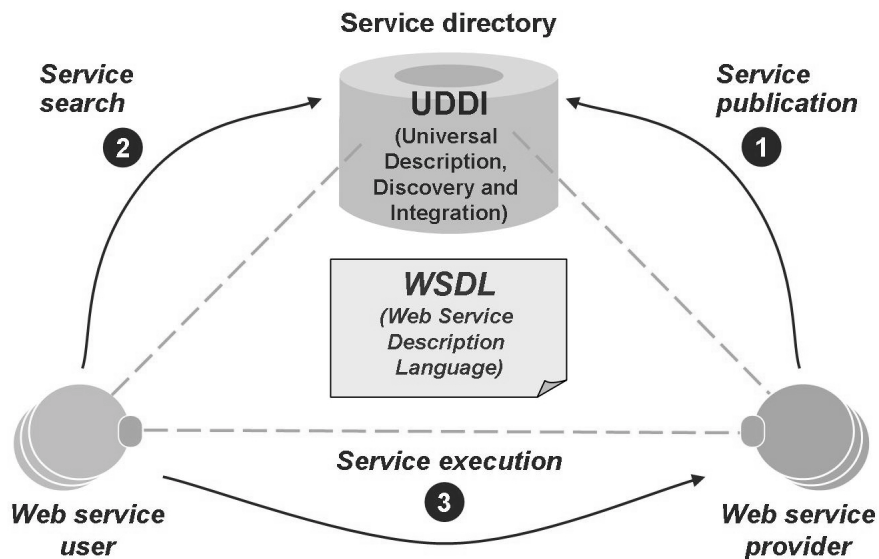


Figure 92: Outline of a Web Service Scenario

Technically, once a Web service has been defined, it can be called up in different ways. For example, it can be called from an ABAP program or from a Business Server Page.

The following outlines how a Web service is created from an RFC-capable ABAP function module.

- The service provider generates the Web service from a function module, for example. A URL and the WSDL file are also generated.
- The service requester creates a proxy object that refers to the URL of the Web service. Next, an ABAP class that matches the proxy object is generated and a logical port is assigned.
- The proxy object is written to and integrated into an executable program (for example, in ABAP) and called up there.



Lesson Summary

You should now be able to:

- Explain Web services
- Describe UDDI and WSDL

Lesson: SAP Business Workflow

Lesson Overview

This lesson provides an overview of the concept and capabilities of the *SAP Business Workflow* (referred to simply as workflow).



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept

Business Example

The leave request process is a good example of how workflow can be used.

SAP Business Workflow Basics

Workflow in SAP systems (or even between SAP systems) aims to increase the speed and transparency of business processes. A workflow model breaks a process down to its individual steps, which are then assigned to various people, or rather, to their roles within the company. The automated sequence of steps ensures that tasks are rapidly assigned to the appropriate employees. This lesson describes two views of the same workflow. One is the view of the people participating in the workflow, the other is a more technical view designed to clarify the process in the system.

A Workflow and its Participants

The process “an employee requests leave of absence” is used as a typical example.

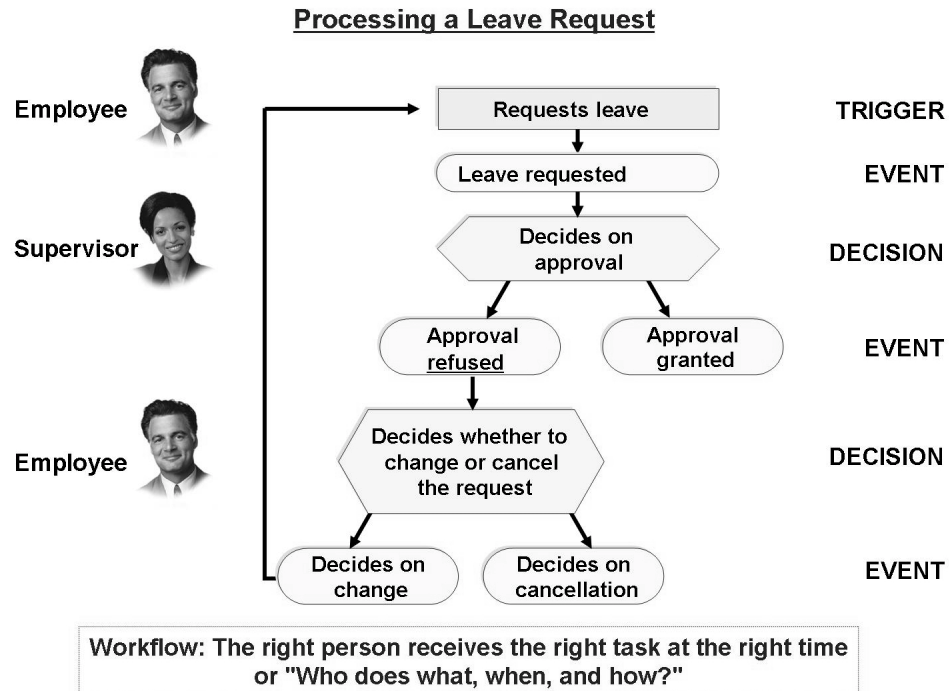


Figure 93: Example of an SAP Business Workflow process

Two persons are involved: the requester and the authorized supervisor assigned to the requester. The requester fills out the form and saves his or her entries. The save action triggers a workflow event, for example, "leave requested". This event is received by an appropriately configured workflow and passed to an approver in accordance with predefined rules. The supervisor (or approver) receives a corresponding **workflow item** in his or her Office Inbox (*Office* → *Workplace* → *Inbox* or transaction SBWP).

A workflow event therefore creates a link between an activity in the SAP system and the people involved. When the approver calls up the workflow item, the approver is automatically referred to the function *Approve/reject request*. There are now two scenarios to consider:

1. The request is approved
2. The request is rejected

If the request is approved (which triggers a workflow event), the requester is informed and the workflow is complete. If the request is rejected, the requester is also informed and has in turn two options:

1. Accept the rejection
2. Change the leave request

If the requester accepts the rejection, the workflow is also complete; if the requester changes the leave request, another workflow item is sent to the approver's Office Inbox.

A Workflow and its Technology

A workflow creates a link between the people who participate in a process and the program steps that belong to this process.



Workflow integration accelerates process flows

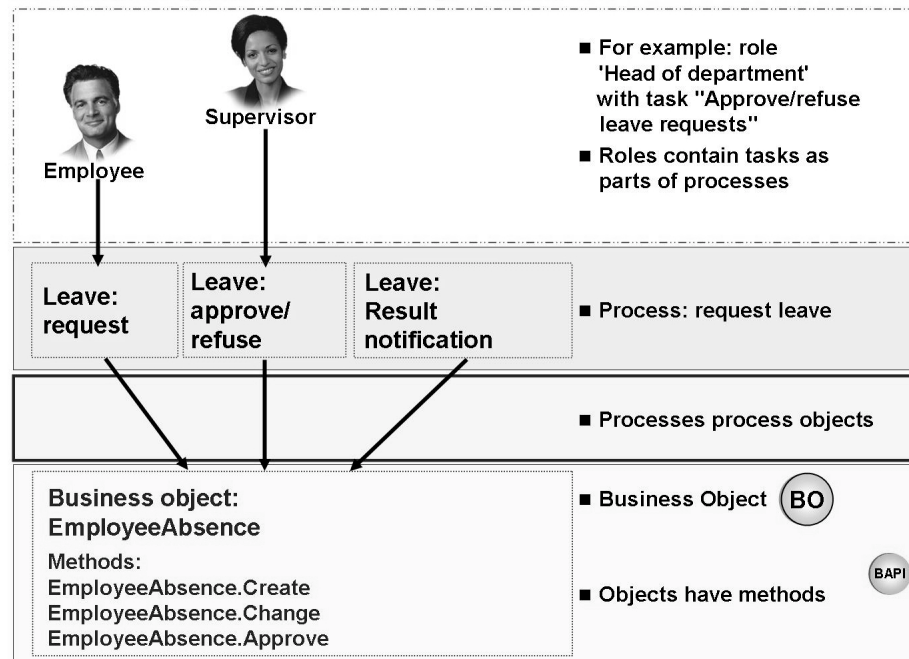


Figure 94: The workflow environment

Different authorizations enable employees in a company to carry out different tasks in one or more systems. Some of these tasks trigger events that are in turn assigned to steps in a workflow model. In other words, when an employee uses his or her authorizations to carry out a task that triggers a workflow event assigned to it, then, for example, a specific method of the relevant business object is called in the system.

An employee's authorizations typically grant access to individual steps in a more comprehensive process. A process works on a particular business object, for example, that is accessed using the methods assigned to it. These methods are defined as BAPIs in the system.

Workflow Application Areas

Since an unlimited number of workflow steps can be assigned to a workflow event, and complex workflow sequences can be assigned to a process, the workflow function is used in a wide variety of business areas in SAP systems. For example, complex process flows in the *SAP CRM* or *SAP SCM* solutions would hardly be possible without workflow integration.



Workflow uses include:

■ Facilitating communication:

- Automatic notification
- Replaces “circulars”



■ Controlling and monitoring simple processes:

- Problem messages
- Notifications for slow system response times, for example



■ Controlling complex processes:

- Approval procedures
- Purchasing using the Internet (such as SAP CRM)

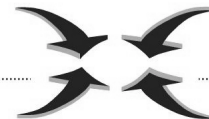


Figure 95: Workflow application areas

Workflow is also particularly well suited to automating the distribution of information on work in progress (for example, information on the status of a purchase order) to all those concerned. Workflow can additionally use generated XML messages to trigger cross-system activities in remote systems. Defining appropriate events and assigning them to business object methods enables you to use workflow in almost any area.

Exercise 12: Leave Request as Workflow

Exercise Objectives

After completing this exercise, you will be able to:

- Describe the *SAP Business Workflow* concept

Business Example

In your production SAP system, the procedures are to be implemented using the *SAP Business Workflow*, hence, you are interested in the concept and possible uses of the workflow.

Task: Management of Leave Requests using SAP Business Workflow


Use *SAP Business Workflow* to generate and process a leave request.

1. Your user SAPTEC-## generates a leave request for today.
2. Check whether the leave request you just entered has arrived in the inbox of the *SAP Business Workplace*.
3. The supervisor approves the leave request.

Solution 12: Leave Request as Workflow



Task: Management of Leave Requests using SAP Business Workflow

Use *SAP Business Workflow* to generate and process a leave request.

1. Your user SAPTEC-## generates a leave request for today.
 - a) Start transaction SWUI_DEMO. In the left navigation area, choose the *Demo for leave request process* entry. Choose *Start*. You see an input template. Please enter *Training* as the department and specify an absence of eight hours; the current date is already entered in the date field but you can still change the entry. Furthermore, enter *Special reasons* as the reason for the absence. Choose pushbutton *Save*. 
2. Check whether the leave request you just entered has arrived in the inbox of the *SAP Business Workplace*.
 - a) Start transaction SBWP. Open the *Inbox* folder. When you double-click the *Workflow* folder, the workflow list of open workflow items is displayed. Your leave request should be displayed there.



Hint: In the context of the demo workflow you are both employee and supervisor in one person. In a real workflow scenario, only your supervisor would see the generated leave request.

3. The supervisor approves the leave request.
 - a) Start transaction SBWP. Open the *Inbox* folder. By double-clicking the *Workflow* folder a list of open workflow items is displayed. Choose your leave request and process it. You start processing using the *Execute* button.  Choose *Approve*. Use the green arrow  or the *F3* button to return to the previous screen. Choose *Close Work Item* to close the workflow.



Hint: In a real workflow the employee would be notified by express mail that his leave request has been approved. This express mail is not sent in the demo workflow.



Lesson Summary

You should now be able to:

- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept

Related Information

For additional information, see the follow-up courses on the SAP Business Workflow:

- BIT 600: *SAP Business Workflow - Concepts*
- BIT 601: *SAP WebFlow – Definition and Use*
- BIT 603: *SAP WorkFlow and Web Scenarios*
- BIT 610: *SAP WorkFlow – Programming*



Unit Summary

You should now be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept
- List various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Make a Remote Function Call
- Explain the evolution from *SAP R/3* to *SAP ERP* and the Enterprise SOA
- Describe the significance of the Web services within the Enterprise SOA
- Explain Web services
- Describe UDDI and WSDL
- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept



Test Your Knowledge

1. Application Link Enabling (ALE) allows you to:
Choose the correct answer(s).
 - ☐ A Exchange data only between SAP systems, as long as they have the same release status
 - ☐ B Exchange data across system boundaries, but only for SAP applications
 - ☐ C Exchange data between collaborating enterprises, using certain formats and technologies
 - ☐ D The communication between different systems of your system landscape
 - ☐ E Update your order data using the appropriate BAPI, only once every 24 hours

2. The following interfaces or communication options are supported by SAP systems:
Choose the correct answer(s).
 - ☐ A HTTP (HyperText Transfer Protocol)
 - ☐ B SMTP (Simple Mail Transfer Protocol)
 - ☐ C RFC (Remote Function Call)
 - ☐ D BAPIs (Business Application Programming Interfaces)
 - ☐ E XDTP (Extended Data Transfer Protocol)
 - ☐ F STP (SAP Transfer Protocol)

3. You can use BAPIs to:
Choose the correct answer(s).
 - ☐ A Request data from an SAP system
 - ☐ B Pass data to an SAP system
 - ☐ C Transfer SAP screen images to third-party applications (such as Microsoft Word)
 - ☐ D Access business processes in SAP systems

4. You access BAPIs in SAP systems using an RFC interface.
Determine whether this statement is true or false.
 - ☐ True
 - ☐ False

5. Business _____ Programming Interfaces are specialized _____ modules. They are accessed using the _____ interface. They are created and managed using the _____ Builder.

Fill in the blanks to complete the sentence.

6. *SAP Business WorkFlow* ensures that:

Choose the correct answer(s).

- ☐ A Appropriately configured business processes can be partially automated
- ☐ B Appropriately configured business processes are executed in consistent sequences
- ☐ C The right employee receives the right work at the right time
- ☐ D Your workflow-supported business processes are handled more efficiently
- ☐ E All your company processes that have been implemented in ABAP run without errors

7. You can also use *SAP Business Workflow* functions (for example, with XML) to trigger functions in other systems.

Determine whether this statement is true or false.

- ☐ True
- ☐ False



Answers

1. Application Link Enabling (ALE) allows you to:

Answer: C, D

ALE is a very powerful method of exchanging data between systems. These systems may be located within the same company, or they may be distributed between several companies. The data is transferred by RFC in a previously defined format. The transfer type may be synchronous or asynchronous.

2. The following interfaces or communication options are supported by SAP systems:

Answer: A, B, C, D

From an SAP system, you can communicate with other systems using, for example, HTTP, SMTP, RFC, or BAPIs. XDTP and STP do not exist.

3. You can use BAPIs to:

Answer: A, B, D

You can use BAPIs to access business processes in an SAP system and to request and transfer data between systems. GUI functions cannot be transferred to third-party products.

4. You access BAPIs in SAP systems using an RFC interface.

Answer: True

BAPIs are nothing other than special, remote-enabled function modules. They can therefore also be addressed using RFCs.

5. Business Application Programming Interfaces are specialized function modules. They are accessed using the RFC interface. They are created and managed using the Function Builder.

Answer: Application, function, RFC, Function

You can start the Business Object Repository using transaction code BAPI, and the *Function Builder* using transaction code SE37.

6. *SAP Business WorkFlow* ensures that:

Answer: A, B, C, D

Supporting business processes with workflow enables the work steps that belong to these processes to be handled on a partially automated basis. These steps are assigned in consistent sequences to the appropriate employee at the right point in time. This enhances processing efficiency. Using workflow obviously does not guarantee that the programs you write will always run without errors.

7. You can also use *SAP Business Workflow* functions (for example, with XML) to trigger functions in other systems.

Answer: True

You can use workflow to send XML messages that then trigger subsequent actions in other systems, as long as they are appropriately configured.

Unit 6

Working with SAP Solution Manager

Unit Overview

SAP Solution Manager supports you during the entire life cycle of your solutions – from the Business Blueprint through configuration to production operation. It provides central access to tools, methods, and preconfigured content, which you can use during evaluation and implementation as well as during production operation of your systems.

This unit provides a detailed introduction to the use of SAP Solution Manager in production operation.

The first lesson in this unit provides an overview of the functions of SAP Solution Manager. The second lesson then discusses the steps required to map the customer system landscape in SAP Solution Manager. These steps are required, for example, to set up central monitoring.



Unit Objectives

After completing this unit, you will be able to:

- Describe the added value of the SAP Solution Manager in an SAP system landscape
- Describe the role of the System Landscape Directory when mapping system landscapes
- List and explain the steps required to connect SAP Solution Manager to a System Landscape Directory
- Create logical components in SAP Solution Manager and assign systems
- Create RFC destinations in SAP Solution Manager for the component systems

Unit Contents

Lesson: Concept of the SAP Solution Manager	294
Lesson: Connecting ABAP-Based Systems to SAP Solution Manager	305
Exercise 13: Connecting ABAP-Based Systems to SAP Solution Manager	327

Lesson: Concept of the SAP Solution Manager

Lesson Overview

The SAP Solution Manager is a tool that provides the administrator with a large number of functions for administering a complex system landscape. This lesson provides an overview of these functions.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the added value of the SAP Solution Manager in an SAP system landscape

Business Example

The SAP Solution Manager provides a central service and support portal and therefore, supports the administrator in his or her daily work.

Introduction

The SAP Solution Manager is a service and support platform for the implementation and operation of SAP systems. It provides content, tools, and procedures for implementing, operating and supporting SAP systems.

The SAP Solution Manager...

- Increases the reliability of your SAP systems
- Reduces the Total Cost of Ownership of your SAP solutions
- Increases the return on investment provided by your SAP solutions
- Is included in the maintenance charges for your SAP systems

The SAP Solution Manager supports customers at the start of the project (to implement an SAP application), during functional and technical implementation, during running operation, and during the optimization of their system landscapes. The SAP Solution Manager also makes it possible to communicate with SAP and its partners at any time and therefore, enhances the extensive on-site service.



Figure 96: What Can the SAP Solution Manager Do?

The SAP Solution Manager is the central implementation and operation platform for SAP applications. To be able to use this central platform, first maintain the systems in your solution landscape in the Solution Manager System Landscape (transaction SMSY), see also SAP Tutors on the SAP Service Marketplace under the Quick Link *rkt-solman*.

Connecting your systems to the SAP Solution Manager then provides you with central distribution and synchronization of Customizing, central test management, monitoring, and incident management, for example. For clearer monitoring and documentation, the systems are combined into Solution Landscapes.

Your core business processes, together with the supporting technical components are documented in these Solution Landscapes. The business processes of the SAP Solution Manager are maintained centrally in the Solution Directory (transaction SOLMAN_DIRECTORY), to which all SAP Solution Manager functions have access. For monitoring, you can, however, also create Solution Landscapes where systems are exclusively monitored on the basis of Early Watch alerts (EWAs) or the CCMS. A system can be part of multiple landscapes.

The SAP Solution Manager provides many functions for the implementation and operation of your SAP Business Suite application(s).



Life -Cycle Approach for Implementation and Operation of SAP Solutions

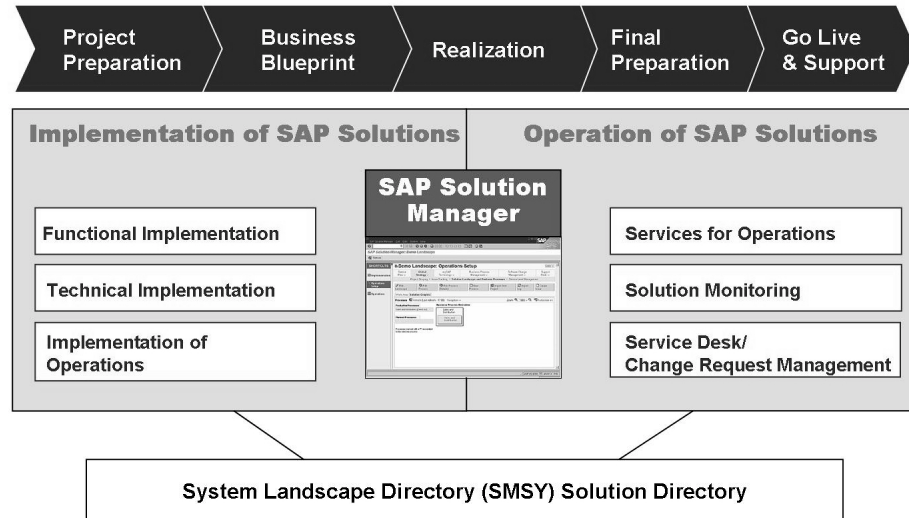


Figure 97: Functions of the SAP Solution Manager

Some Functions of the SAP Solution Manager

- Preventative services: EarlyWatch Alert, GoingLive Check, and GoingLive and Functional Upgrade Check.
- Continuous Improvement Services: Solution Management Review Service (SMR) and Solution Management Optimization Services (SMO)
- Best practices for your SAP application:
Documents and services that are based on SAP's experience from production customer installations.
- Application and System Monitoring: Technical monitoring of your SAP applications including the availability of interfaces, individual system components, business process monitoring, service level and a graphical alert monitor.
- SAP Service Desk: Central message management for all SAP applications (including attached files or error contexts), interface to the SAP Service Marketplace, SAP Notes database search and automatic import of SAP Notes (Note Assistant), and the customer solution database for customer-specific questions and answers.
- SAP Remote Support: Safer remote access using Internet connections

Installation of the SAP Solution Manager

SAP Solution Manager is installed as a separate SAP system. SAP Solution Manager 4.0 is based on SAP NetWeaver 7.0. For an overview of the release strategy, see the SAP Service Marketplace (<http://service.sap.com/solutionmanager> → *Release Strategy*).

The SAP Solution Manager is available free of charge as part of your maintenance contract. You can order the installation through the SAP Service Marketplace (<http://service.sap.com/solutionmanager> → *Order Information*).

Areas of Operation of the SAP Solution Manager

The SAP Solution Manager provides many functions for implementing and operating SAP applications. During implementation, project activities are primarily supported with project planning tools.

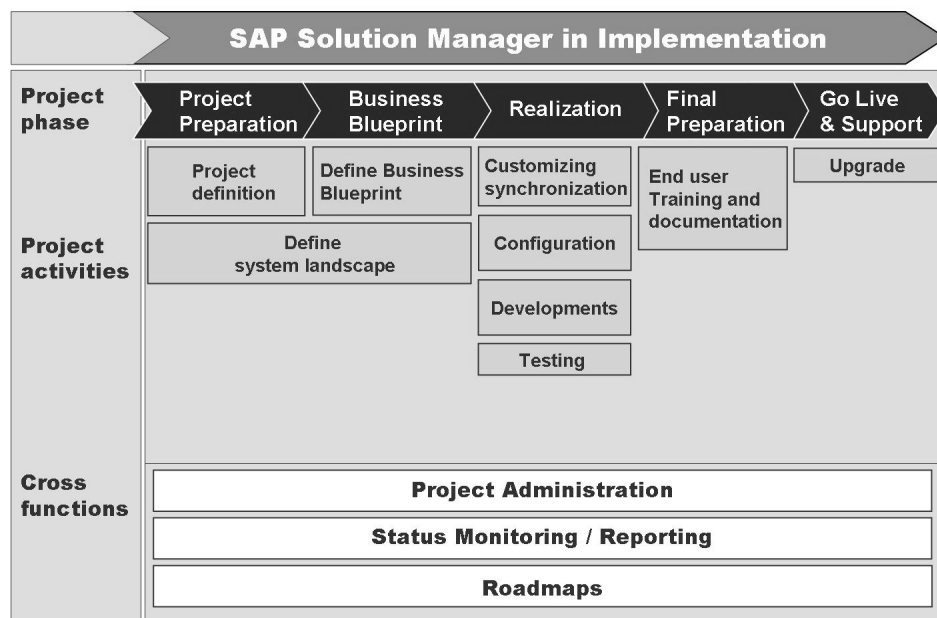


Figure 98: SAP Solution Manager - Implementation Functions

The SAP Solution Manager supports you in performing central activities during the implementation.

- Project management support with which you can manage scheduling periods, employees, and other project information.
- A Business Process Repository, which provides templates for mapping processes. These templates are required to define the project scope and the central enterprise processes.
- Tools for implementing the integrated business scenarios.
- Integrated Implementation Guides that support you during the Customizing activities for the business processes.

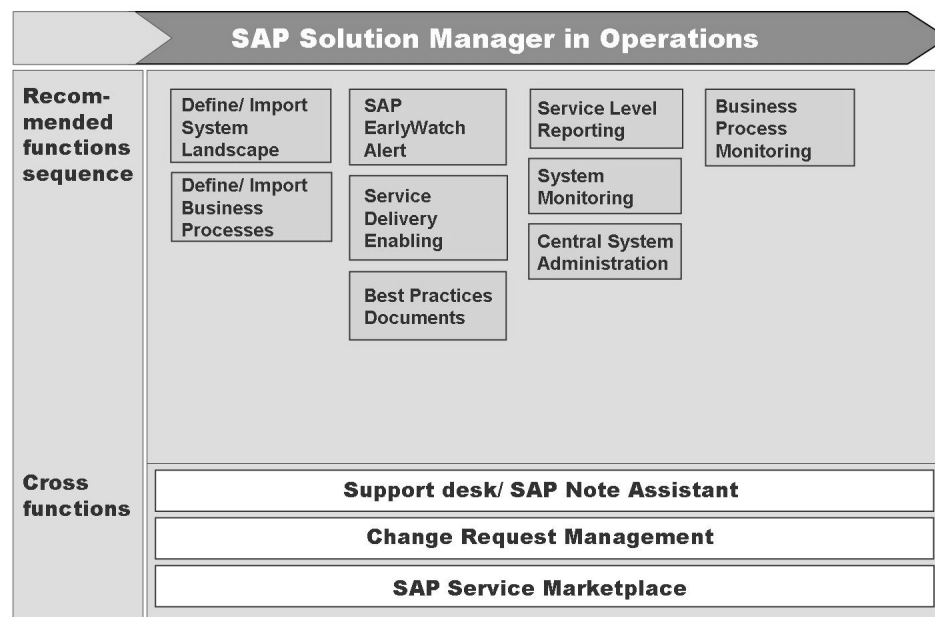


Figure 99: SAP Solution Manager - Operation Functions

- Support when monitoring business processes
- Central system monitoring
- Possibilities for central administration tasks
- Best Practices are provided
- Remote Services, such as the SAP EarlyWatch Alert Service and others

You can easily order services using the SAP Solution Manager. The process of an order is outlined in the following figure.

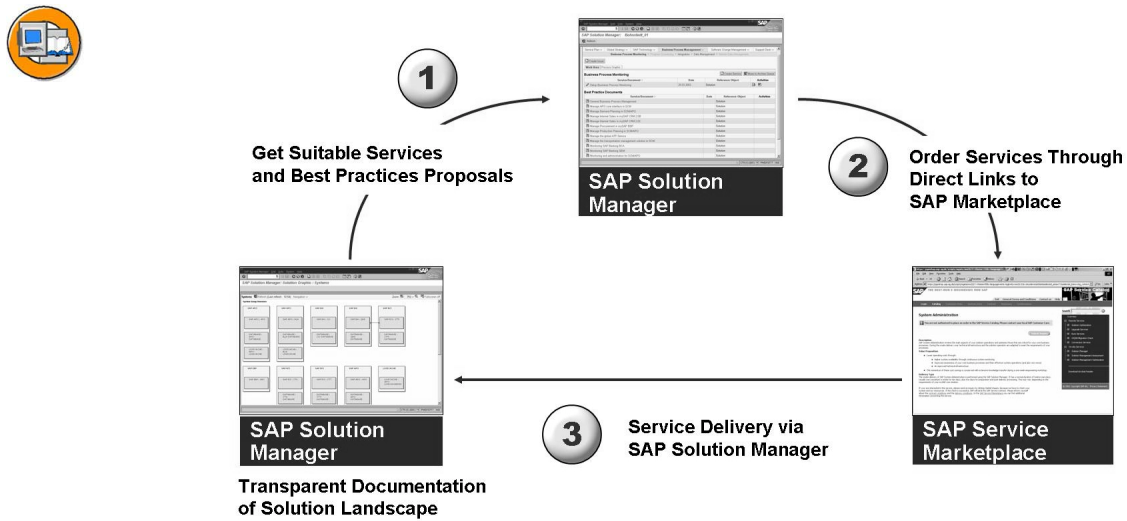


Figure 100: Service Suggestions and Service Ordering

The SAP Solution Manager can provide you with suggestions for services which, depending on your system landscape, are useful for your system environment.

Monitoring with the SAP Solution Manager

The SAP Solution Manager provides many options for monitoring your SAP systems (that are connected to the SAP Solution Manager). You can not only monitor individual systems, but also check the run times of cross-system business processes. Many reporting functions are also provided.

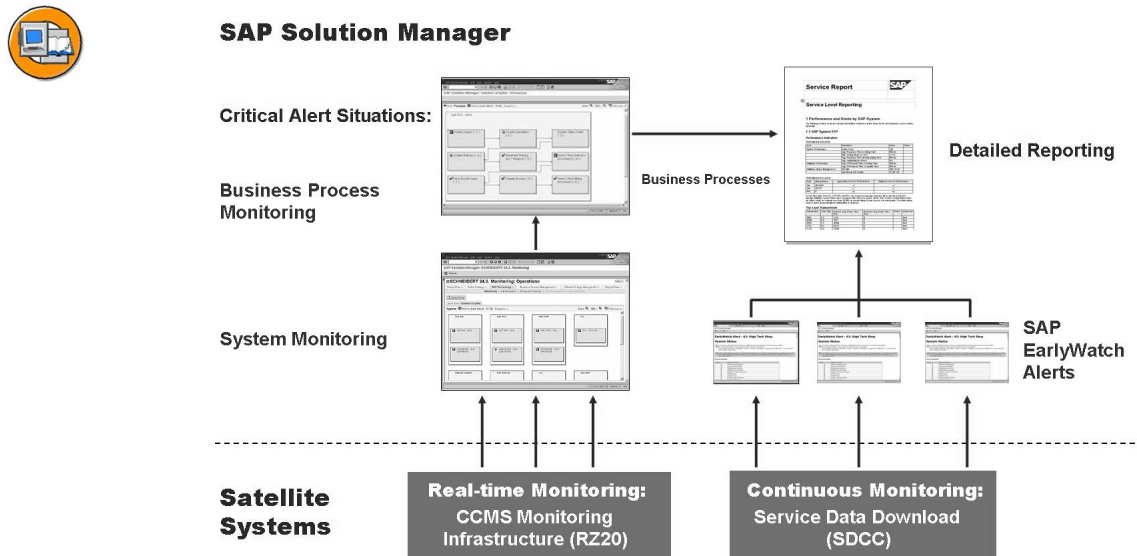


Figure 101: System Monitoring with the SAP Solution Manager

The monitoring architecture of the SAP Solution Manager is outlined in the figure. The satellite systems automatically collect the actual data. The SAP Solution Manager collects this data, aggregates it, evaluates it, and displays the result graphically. CCMS Monitoring (RZ20) and the Service Data Control Center (SDCC) act as data collectors on the satellite system side.

The SAP Solution Manager's system monitoring uses the data from CCMS monitoring and displays this in a business process-oriented context. You can set up central monitoring of all components of your SAP systems relevant to operation here, as in CCMS monitoring. You can also use predefined alert messages, which react to customizable threshold values.

The business process monitoring allows your departments to obtain a quick overview of the technical status of your business processes. CCMS and other alerts are displayed graphically and oriented by business process. This graphical display helps you to quickly identify the process where a problem has occurred. Detailed information about the problem, and the possibility of escalating the problem using a Service Desk message are linked to the warning message.

You can define system status reports yourself using Service Level Management. These include the EarlyWatch Alert or the Service Level Report, which automatically provide you with regular reports (in HTML or Microsoft Word format) about the current status of your SAP application. You only need to schedule these services once. They then collect data about your systems regularly, aggregate it, and evaluate it.

The Service Desk of the SAP Solution Manager

The Service Desk of the SAP Solution Manager provides a complete IT infrastructure for providing back office support with message handling for the SAP applications in your company. The SAP Solution Manager acts as a central collection point for all the support messages that are created in your SAP systems (see the following graphic).

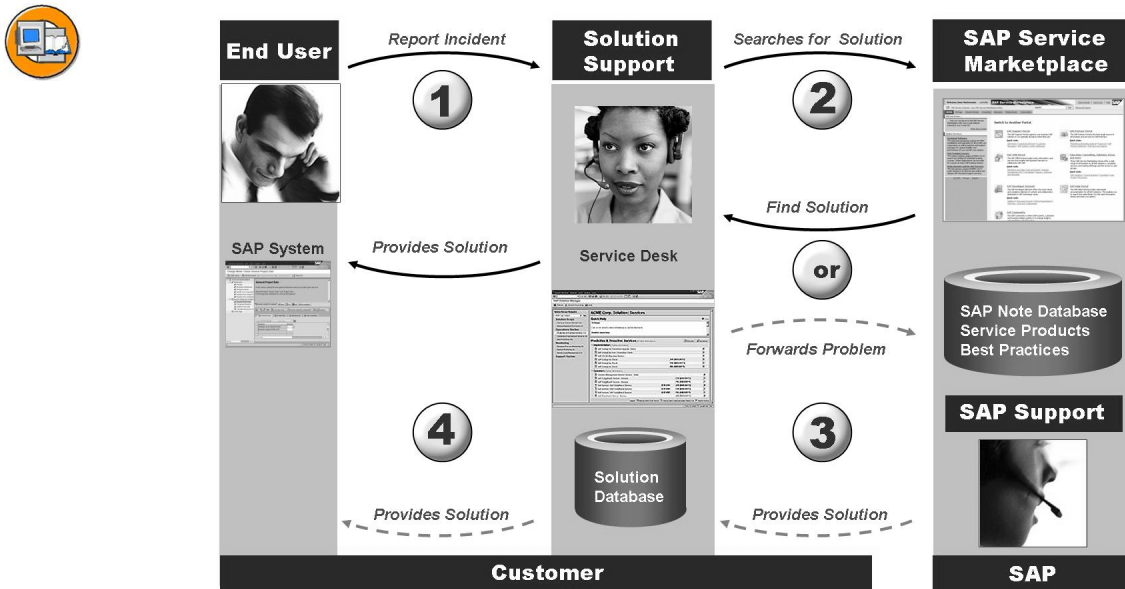


Figure 102: End User Message Creation

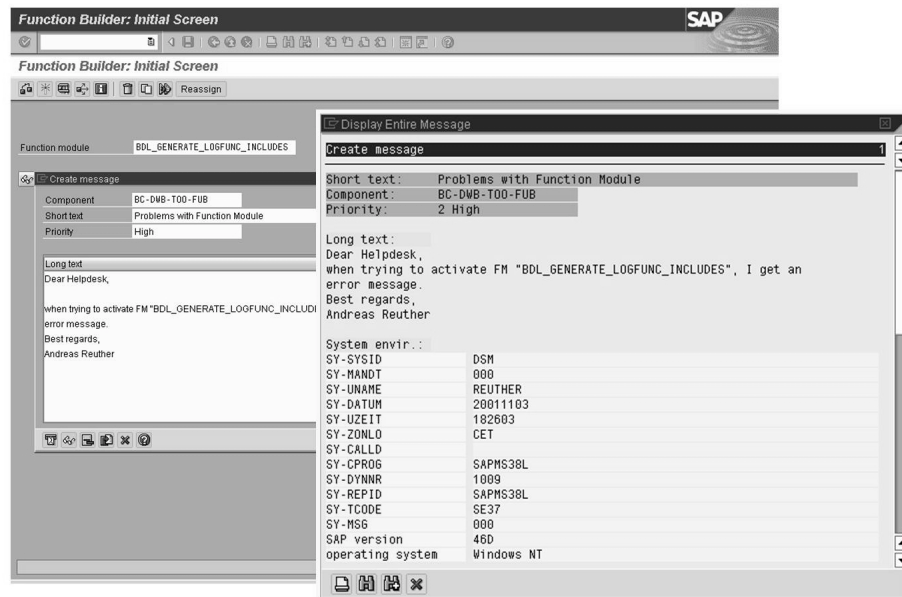


Figure 103: SAP Service Desk

All end users can send a message to your back office from any SAP transaction by choosing *Help* → *Create Support Message*. When they do so, important system data is automatically attached to the message, which significantly speeds up error identification and message processing. This message is sent directly to the SAP Solution Manager using an RFC connection. It is possible to sort the message using components and to assign an urgency level to the message. External files can, of course, be included.

The back office department can view and process the messages created by the end users in the SAP Solution Manager. To solve any end user problems that may occur, support has direct access to all SAP Notes, their own customer solutions database and other documents and tools that are provided by SAP on the SAP Service Marketplace. After a successful search for a problem solution, support can send the message back to the user with a description of the solution. In urgent cases or for very special problems, support can forward the error message to SAP through the SAP Service Marketplace.

Change Request Management is available as of SAP Solution Manager 3.2. This provides many functions to control development and Customizing activities in your SAP systems.



- Manage all change requests
- Classify change requests
- Approver workflow
- Status tracing
- Complete change history



Lesson Summary

You should now be able to:

- Describe the added value of the SAP Solution Manager in an SAP system landscape

Related Information

- On the SAP Service Marketplace under <http://service.sap.com/solutionmanager>

Lesson: Connecting ABAP-Based Systems to SAP Solution Manager

Lesson Overview

Before you can use SAP Solution Manager for change request management or central system monitoring, you need to perform certain configuration steps in the SAP Solution Manager system and in the component systems that have been connected.

Following a brief explanation of the purpose of a System Landscape Directory (SLD), this lesson begins with an introduction to the SLD. Information about the current system landscape is available to the SLD. To enable this data to also be available in SAP Solution Manager, you can establish regular data exchanges between the SAP Solution Manager system and the SLD. The necessary steps are discussed in this lesson.

At the end of this lesson, logical components are created in the SAP Solution Manager system and systems are assigned. Furthermore, RFC connections are established from SAP Solution Manager to the component systems. SAP Solution Manager can use these RFC connections in central monitoring or in change request management.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the role of the System Landscape Directory when mapping system landscapes
- List and explain the steps required to connect SAP Solution Manager to a System Landscape Directory
- Create logical components in SAP Solution Manager and assign systems
- Create RFC destinations in SAP Solution Manager for the component systems

Business Example

Your company wants to use SAP Solution Manager for change request management and central system monitoring. As the system administrator, it is your responsibility to map your company's existing SAP system landscape in the SAP Solution Manager system as well as establish RFC connections from SAP Solution Manager to the component systems that have been connected.

Mapping the System Landscape in SAP Solution Manager

If you want to use SAP Solution Manager in implementation and template projects, when tracking change requests (change request management), in Customizing synchronization (Customizing Scout and Customizing Distribution), and in production operation (Solution Monitoring, Services, and Service Desk), you must create and manage your existing system landscape locally in SAP Solution Manager. To be able to do this, all of the products and the associated system landscapes must be made known in SAP Solution Manager.



System landscape definition:

- For each project, all products and associated DEV, QAS, and PRD systems must be defined in Project Administration (SOLAR_PROJECT_ADMIN).
- Before a project landscape can be defined, products, systems, and associated RFC destinations must be created in transaction SMSY (System Landscape Maintenance).

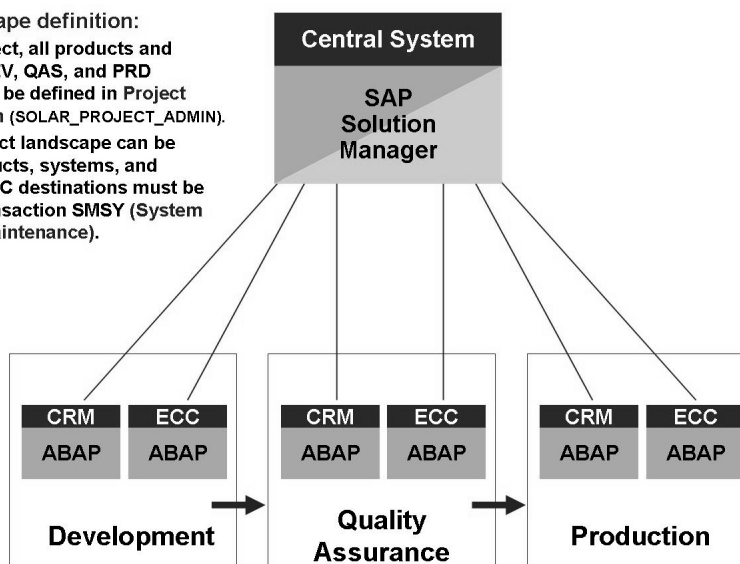
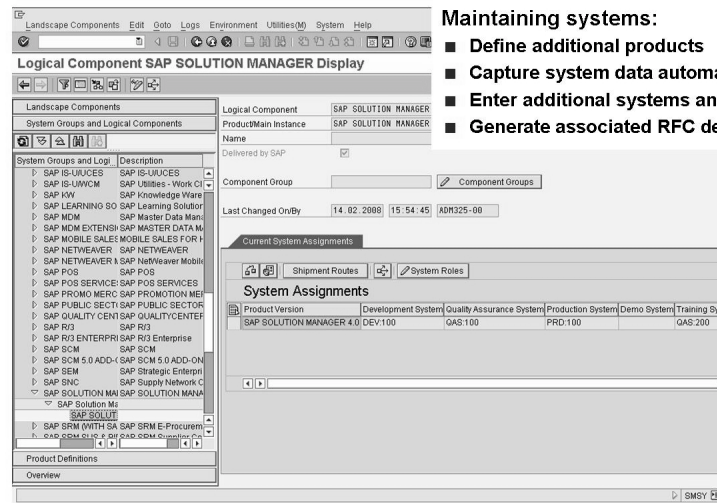


Figure 104: Definition of System Landscape in SAP Solution Manager

The system landscape defined in SAP Solution Manager is the basis for further work with SAP Solution Manager.

Transaction SMSY

The current system landscape is defined in SAP Solution Manager in transaction SMSY (*System Landscape – SAP Solution Manager*).



Maintaining systems:

- Define additional products
- Capture system data automatically or manually
- Enter additional systems and basic data
- Generate associated RFC destinations



- An SLD can be used in the background to capture system data automatically.
- RFC destinations need to be created for each logical system (system: client).

Figure 105: Transaction SMSY

Transaction SMSY can be used to provide the following functions, for example, in the SAP Solution Manager system:

- Definition of non-SAP products for further use in system landscape maintenance
- Manual data entry, for example, for servers, non-ABAP systems, and planned systems
- Generation of RFC destinations for the component systems; any errors that occur with RFC connections are recorded in a log.

Landscape components (servers, databases, systems, and system components) can be created manually in transaction SMSY or automatically through a periodic comparison with the System Landscape Directory being used. In this case (recommended by SAP), the SLD supports you and simplifies data maintenance for your system landscape in SAP Solution Manager.

Procedure for Configuring SAP Solution Manager

For information about the procedure for configuring SAP Solution Manager, refer to the *SAP Reference IMG*, which you can call from transaction SPRO.

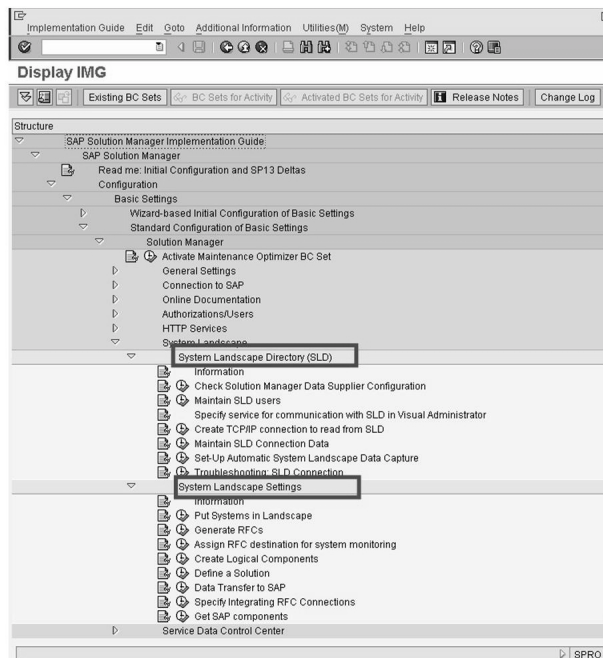


Figure 106: Implementation Guide: Procedure for Configuring the System Landscape

Here, you must distinguish between basic settings and scenario-specific settings. To be able to work with SAP Solution Manager, you must first make all of the basic settings before you can configure the scenario-specific functions. Example: For issue management (problem tracking), you must have made the basic settings for business partners and number ranges because these are also required for issue management.

Addendum: Logical Systems

One of the first steps when configuring an SAP system based on AS ABAP is to name the logical systems. This is necessary because the SAP systems are referenced when logical system IDs are used to exchange messages. In an SAP context, a logical system corresponds to a client. As a result, you must first create logical system names. For this, you use transaction BD54. The logical system definition can be recorded in a transport request and transported to other systems.



A logical system corresponds to a client in an SAP system.

Creating a Logical System (BD54)

Assigning a Logical System to a Client (SCC4)

Figure 107: Addendum: Creating Logical Systems and Assigning Clients

Transaction SCC4 is used to assign a logical system to a special SAP system client. Only one logical system can be assigned to each client. Furthermore, the logical system must be unique throughout the company.

The System Landscape Directory (SLD)

The System Landscape Directory (SLD) is the central directory of all system landscape information relevant for software lifecycle management. After the installation, the SLD is available on each SAP system based on AS Java (since SAP Web AS 6.40). However, it must be configured before use.

Purpose

Today, system landscapes comprise several hardware and software components that depend on each other in terms of their installation, software updates, and interface requirements.

Therefore, to simplify system landscape management, you require an overall concept that comprises aspects of implementation, upgrade, and maintenance. This is where the System Landscape Directory comes into play. The SLD simplifies system landscape management. Its content is based on the Common Information Model (CIM) standard.



Hint: The CIM standard is a general schema for describing elements of a system landscape.

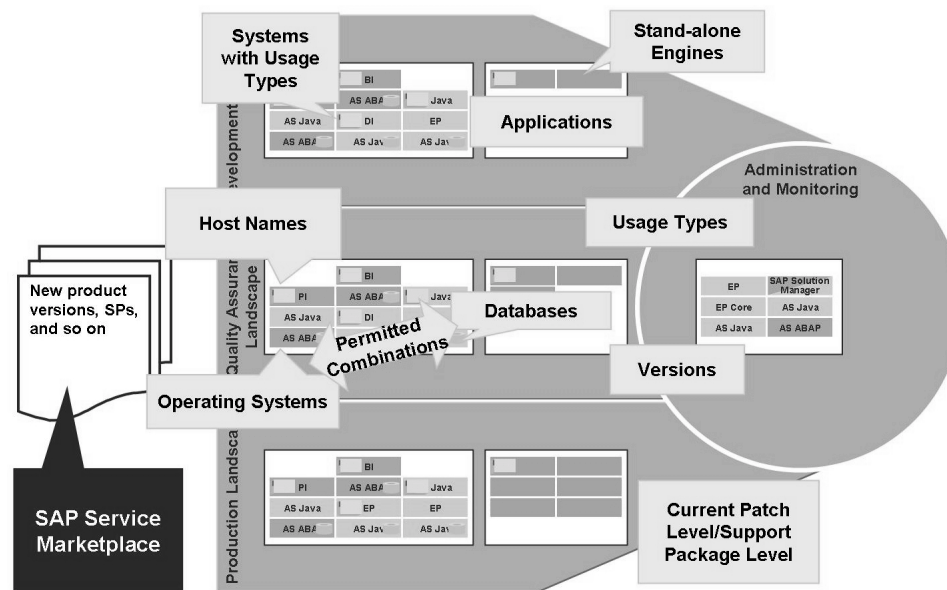


Figure 108: System Landscape Directory – Purpose

SAP has a master component repository that contains the latest information about all available SAP products. This information includes product versions, Support Package levels, and their dependencies. The content of this master component repository is published on SAP Service Marketplace, so that customers can update their component information in their SLD. For more information, see SAP Note 669669 – *Updating the SAP Component Repository*.

Customers can also supplement their component information in the SLD with additional information about the third-party products that they use. While the SLD contains information about all of the products and components that the customer has installed, the landscape description in the SLD is an accurate representation of the current customer landscape.

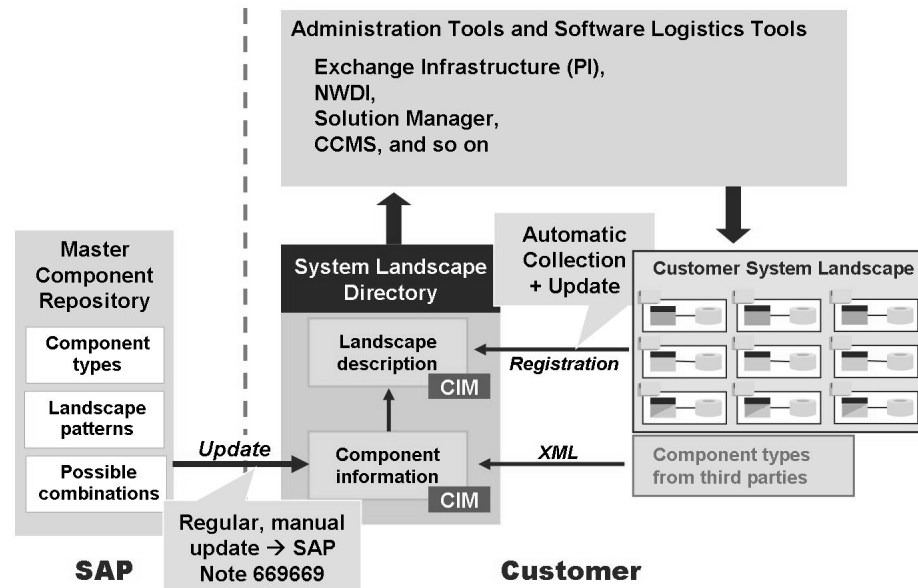


Figure 109: Using the SLD for Landscape Management

Applications and tools (for example, for installation or administrative purposes) can access the information provided by the SLD. The SLD acts as the central provider of information for the entire system landscape.

Connecting ABAP-Based SAP Systems to the SLD

In order for the SLD to receive the data automatically sent by the SAP systems that have been connected, you must start and configure an SLD bridge. The SLD bridge converts incoming data from the data supplier into a CIM-compatible format. In this context, RFCs are used to exchange data with ABAP-based systems. As a result, an SAP Gateway must be configured and the SLD bridge then restarted.

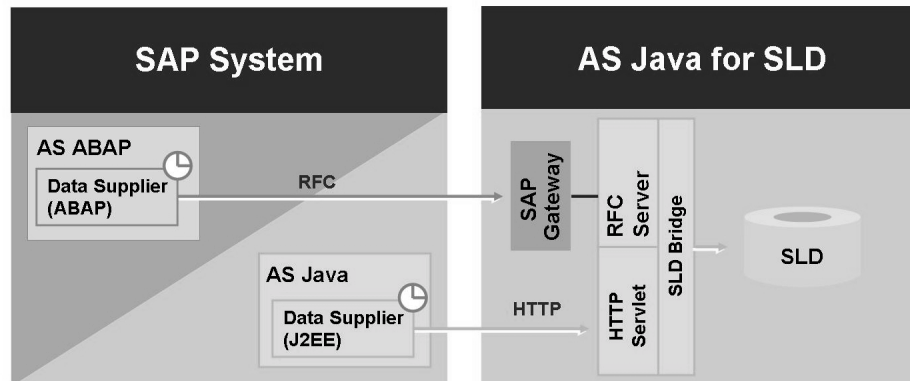
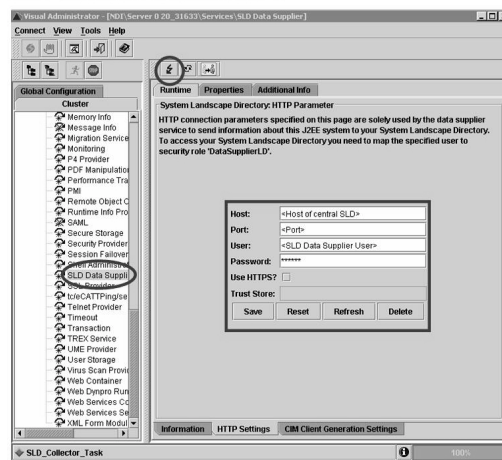


Figure 110: SLD Data Suppliers

To enable SAP systems to automatically send their system data to the SLD, you must configure data suppliers in these systems. For ABAP-based systems, this can be done using transaction RZ70, which uses an RFC connection to the SLD bridge. Java-based programs can use an HTTP connection to transfer system information to the SLD server. The *Visual Administrator* is used for configuration purposes .

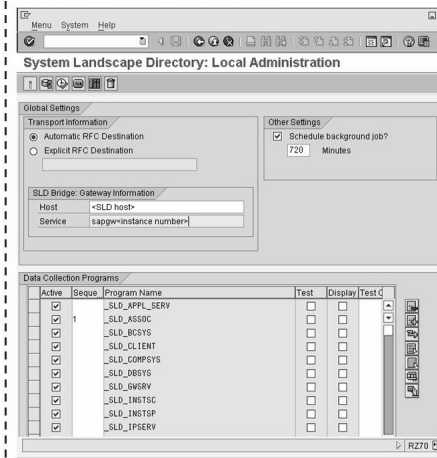


SAP Systems with AS Java



Visual Administrator:
Server → Services → SLD Data Supplier

SAP Systems with AS ABAP



Transaction RZ70

Figure 111: Manual Registration of SAP Systems versus SLD



Hint: During installation, SAP systems can be connected with an SLD that already exists. In the case of AS ABAP, transaction RZ70 and the RFC destination used (*SLD_UC* for a Unicode target system or *SLD_NUC* for a non-Unicode target system) are automatically configured here. In the case of AS Java, the setting in the Visual Administrator is automatically configured here.



Hint: Following a successful configuration in transaction RZ70, two background jobs are scheduled: The job *SAP_SLD_DATA_COLLECT* runs periodically and the job *SAP_SLD_DATA_COLLECT_STARTUP* runs each time the system starts. These jobs ensure that the system information in the SLD is automatically up-to-date.

The Web UI of the SLD

The System Landscape Directory has a user interface that can be accessed using the URL <http://<SLD-Host>:<Port>/sld>.

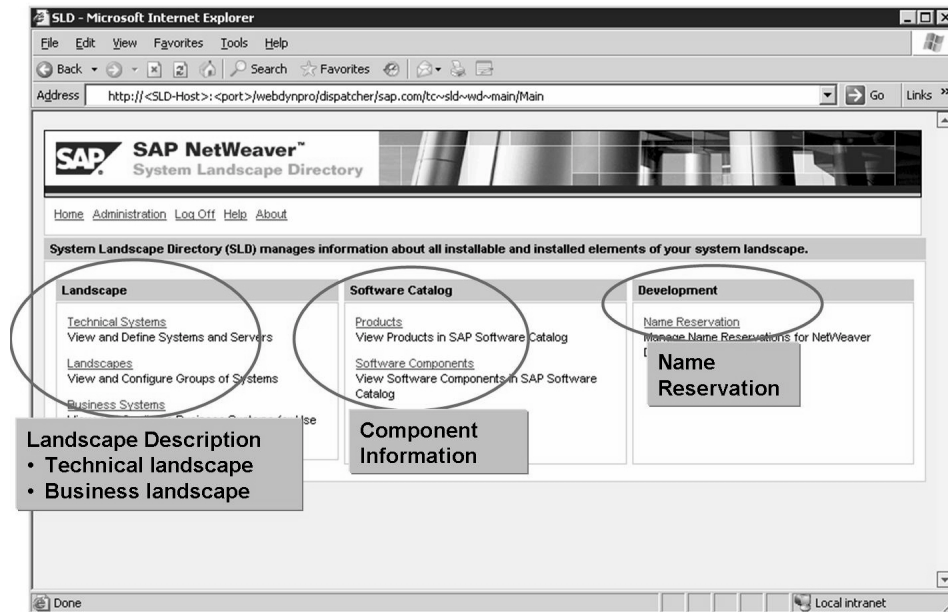


Figure 112: SLD – Start Page

From the SLD start page, you can access the following three areas:

- The description of the current system landscape (that is, the systems that have been installed and their software components)
- The directory of all software components that are available and, in theory, can be installed (both the software components delivered by SAP and customer-defined software components)
- A name reservation service (name server) that enables users to create unique names for software objects



Note: For more information about the System Landscape Directory, see the *Planning Guide – System Landscape Directory* on SAP Service Marketplace, quick link `/instguidesnw70` in the area *Installation* → *1 – Planning*.

Connecting the SAP Solution Manager System to the System Landscape Directory (SLD)

Before you can use SAP Solution Manager for change request management or central monitoring, all of the systems required here (for example, the development, quality assurance, and production systems in the case of change management) must be maintained in transaction SMSY in the SAP Solution Manager system. Only then are these systems and their roles known to the SAP Solution Manager system.

In principle, you can maintain the system landscape in SAP Solution Manager manually. However, if you have already connected all of your SAP systems to a System Landscape Directory, you can automatically transfer the system landscape data from the SLD to your SAP Solution Manager system. An SAP Java Connector (JCo) is used to transfer this data.. The figure below uses a roadmap to illustrate the necessary steps.

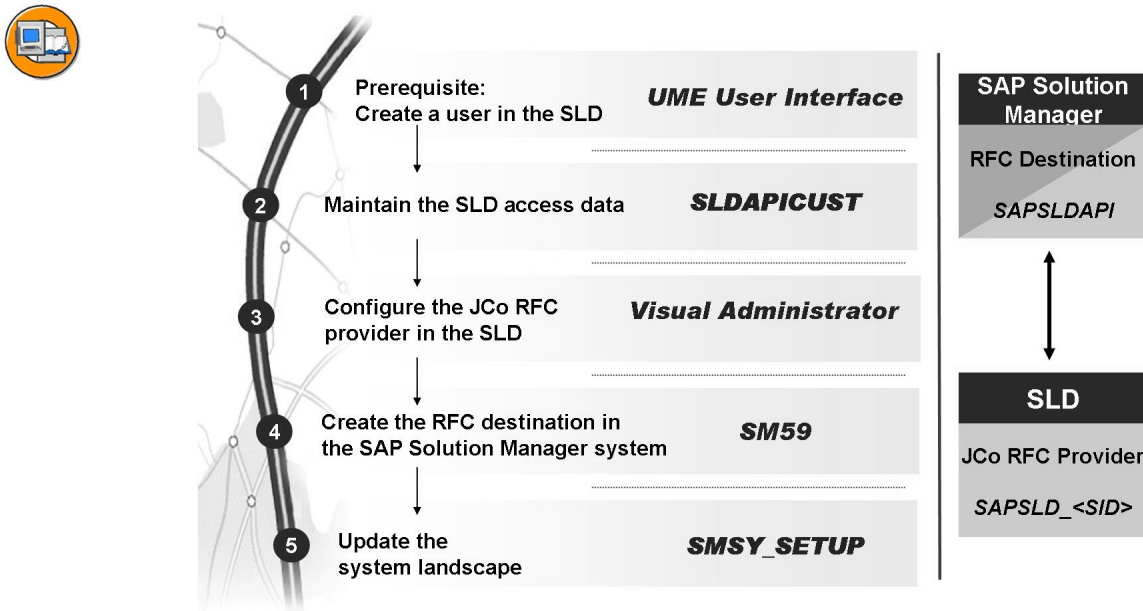


Figure 113: Roadmap: Transferring Data from the SLD to SAP Solution Manager

The following data can be automatically transferred from the SLD to the system landscape of SAP Solution Manager and then automatically updated on a regular basis:

- For systems based on AS ABAP: System name, message server, clients, installed software components, release, Support Package level, instances and assigned servers, server data, databases, and database servers
- For systems based on AS Java: Instances and assigned servers, server roles (types), installed software components, release, Support Package level



Hint: You should connect the System Landscape Directory to SAP Solution Manager, especially if you want to use SAP Solution Manager Diagnostics (SMD) to monitor systems based on AS Java.

Creating a User to Communicate with the SAP Solution Manager System in the SLD

To enable you to work with the SLD, you must first install and configure the SLD (see also SAP Note 764393 – *Configuration of the SAP System Landscape Directory*). You also require the relevant authorizations for the SLD.

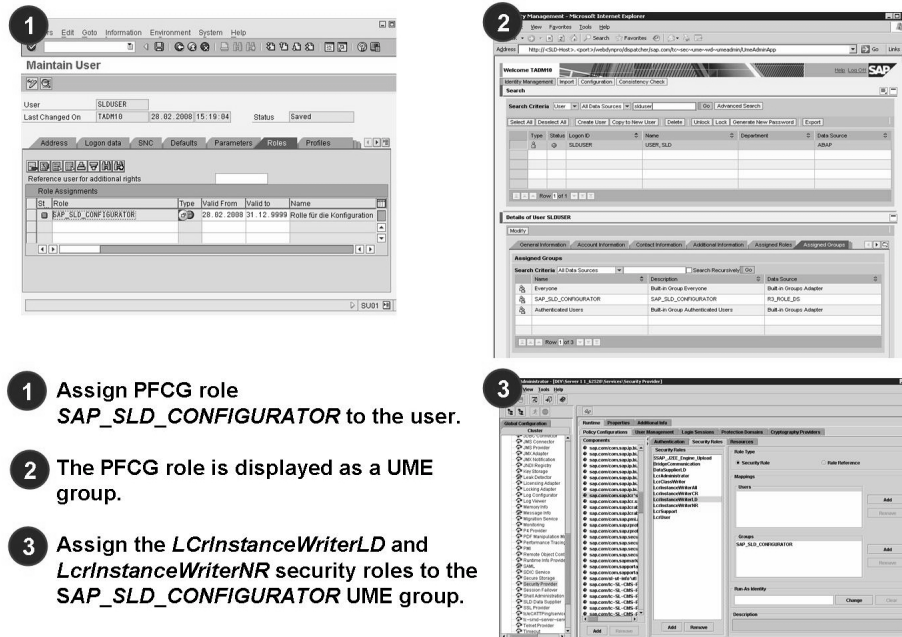
The SLD functions are protected against unauthorized access. Here, the SLD uses J2EE security roles and their associated UME actions (for example, the UME action *com.sap.lcr.LCrInstanceWriterLD* is assigned to the J2EE security role *LCrInstanceWriterLD*).

These J2EE security roles and UME actions must be assigned to the individual users or user groups before the SLD can be used. Therefore, the J2EE security role *LCrInstanceWriterLD*, for example, should be assigned to the UME group *SAP_SLD_CONFIGURATOR*. In this way, users assigned to the UME group *SAP_SLD_CONFIGURATOR* automatically obtain the J2EE security role *LCrInstanceWriterLD* and, as a result, have change authorization in the SLD for the landscape description area.



Note: For detailed information about setting up the SLD, see the *Post-Installation Guide – SLD of SAP NetWeaver 7.0* on SAP Service Marketplace, quick link [/instguidesnw70](#) in the area *Installation* → 5 - *Configuration*.

An SLD user with administration authorization is required for data exchanges between the SAP Solution Manager system and the SLD. The following figure shows how to create this user if the SLD user data is within the ABAP schema of an SAP system.



- 1 Assign PFCG role **SAP_SLD_CONFIGURATOR** to the user.
- 2 The PFCG role is displayed as a UME group.
- 3 Assign the **LCrInstanceWriterLD** and **LCrInstanceWriterNR** security roles to the **SAP_SLD_CONFIGURATOR** UME group.

Figure 114: Creating a User in the SLD

In this case, you first use transaction SU01 to create a user and then assign the PFCG role **SAP_SLD_CONFIGURATOR** to this user (step 1 in the figure). Since PFCG roles are shown as UME groups (step 2), it is sufficient to use the Visual Administrator to assign the J2EE security roles **LCrInstanceWriterLD** and **LCrInstanceWriterNR** to the UME group **SAP_SLD_CONFIGURATOR** (component **sap.com/com.sap.lcr*sld**). In this way, the user created earlier now has change authorization in the SLD for the landscape definition and name reservation areas.

Maintaining SLD Connection Data

An RFC connection is used to connect the SAP Solution Manager system to the System Landscape Directory (additional information is provided later in this lesson). This is done by calling a registered server program that was defined on AS Java in the SLD (additional information is provided later in this lesson). This server program is executed using the HTTP address maintained in transaction **SLDAPICUST** and the user ID defined there.

The user created in the System Landscape Directory is used as the user ID.

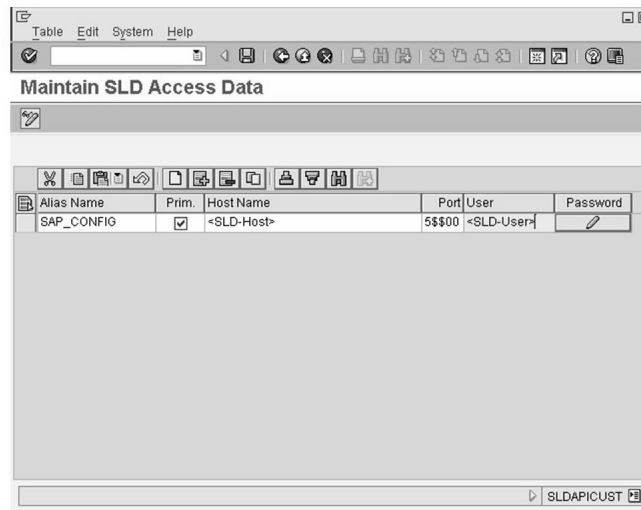


Figure 115: Maintaining the SLD Access Data

Configuring the JCo RFC Provider Service in the SLD

In order to send data from the SLD to AS ABAP in the SAP Solution Manager system, you must start an RFC listener in the SLD. This listener is defined in the *JCo RFC Provider* service area within the Visual Administrator (see the figure below) and waits for ABAP calls.

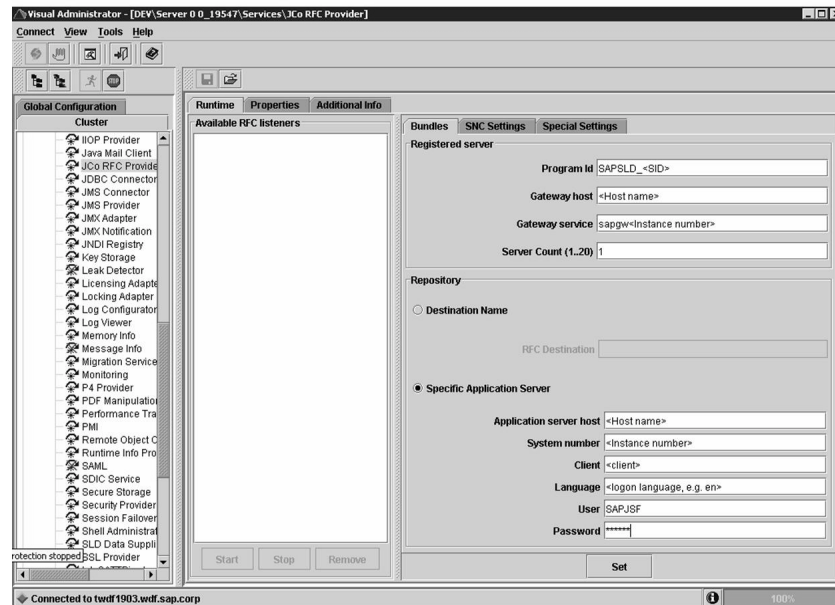


Figure 116: Configuring the SLD for Data Transfers to SAP Solution Manager

You can use any ID as a *Program ID*. The SLD uses this ID to log on to the SAP Solution Manager system (transaction SMGW, menu path: *Goto* → *Logged on Clients*, column: *TP Name*).

Creating an RFC Destination from the SAP Solution Manager System to the SLD

An RFC destination for reading data from the SLD must be created in the SAP Solution Manager system.

Here, the program ID created in the Visual Administrator in the previous step is used to create the RFC destination **SAPSLDAPI** (type **T**) in transaction SM59 as a *Registered Server Program* (see the figure below).



Two RFC destinations are used for the SLD connection:

- The ABAP API uses SAPSLDAPI.
- LCRSAPRFC is necessary for reading the exchange profile (for the PI usage type) – not used in this training course.

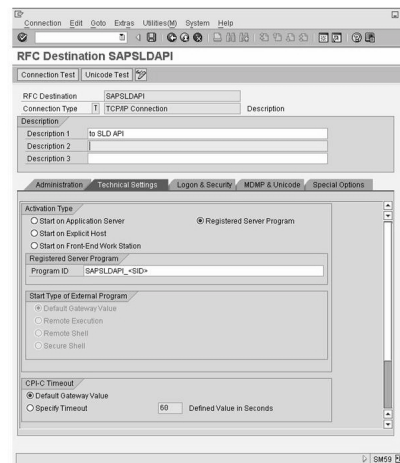


Figure 117: RFC Destination for the SLD



Note: In addition to the RFC destination *SAPSLDAPI*, you also require another RFC destination for the SLD, in association with the SAP Exchange Infrastructure (usage type PI). In this case, you require the RFC destination *LCRSAPRFC*, which is used to read the exchange profile.

You can use transaction SLDCHECK to test the connection to the SLD. This transaction produces a log containing current configuration data, test results or errors, and information about the check.

Scheduling Periodic Data Transfers from the SLD

If a connection has been established between the SAP Solution Manager system and the SLD, you can automate data transfers from the SLD in order to gather system data for the system landscape in SAP Solution Manager. This data includes server, database, and system data.

Transaction SMSY_SETUP is used to schedule periodic data transfers.

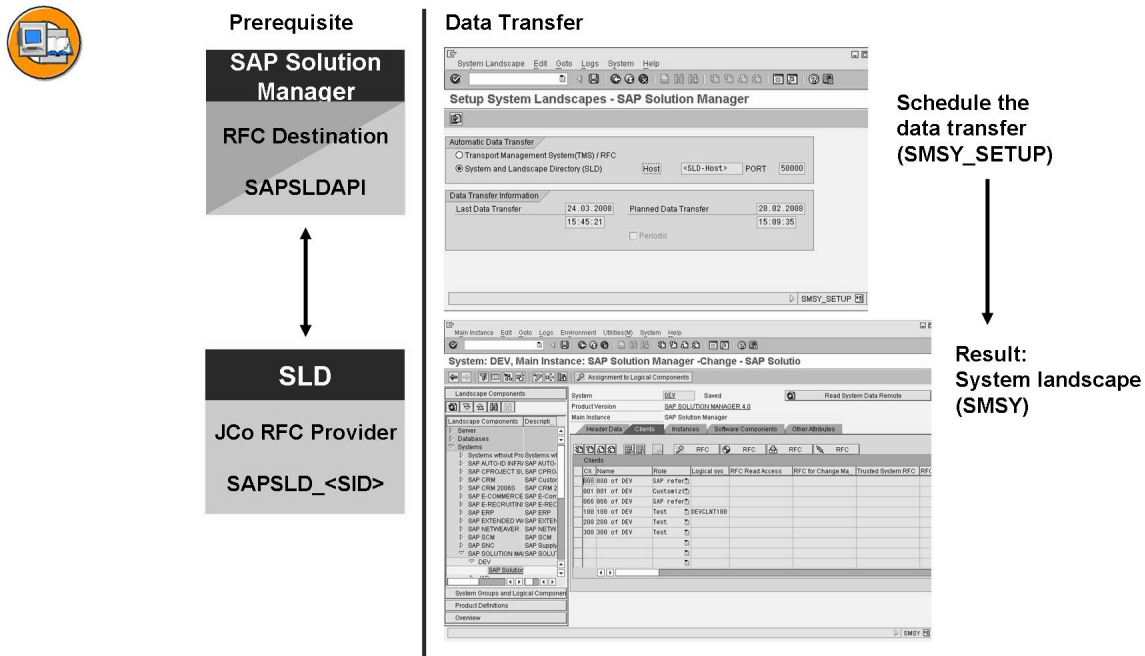


Figure 118: Using Transaction SMSY_SETUP to Update the System Landscape in SAP Solution Manager

In transaction SMSY_SETUP in the SAP Solution Manager system, you may select either the Transport Management System (TMS) or the SLD as the source for the data transfer. SAP recommends that you use the SLD as a data source, if the SLD has been set up in your system landscape. The TMS retrieves only system names, clients and software components in the system (for systems in the transport domain). If, in this case, you want to obtain additional data about servers and databases, you need to establish an RFC connection from SAP Solution Manager to the component systems. Irrespective of the transport domain, the SLD retrieves system names, clients, and software components from the systems, databases, and servers.



Hint: Depending on the Support Package level of the SAP Solution Manager system used, the data is read from the transport domain also when you select the *System and Landscape Directory* option.

Transaction SMSY_SETUP is used to schedule the *LANDSCAPE FETCH* job, which is to transfer data from the SLD on a daily basis. This job uses the SLD connection data as defined in transaction SLDAPICUST.

➔ **Note:** For information about troubleshooting the connection between the SAP Solution Manager system and the SLD, refer to the online documentation for SAP Solution Manager in the area *SAP NetWeaver Problem Analysis Guide (PAG) → System Landscape Directory Problem Analysis Scenarios → Access To SLD From ABAP Fails* (in the online documentation, search for the term “SAP NetWeaver Problem Analysis Guide (PAG)” and, from there, choose the relevant link to navigate to the PAG).

Addendum: Manually Adding Systems to SAP Solution Manager

For SAP products connected to the System Landscape Directory, the SAP Solution Manager system can automatically determine displayed data in transaction SMSY (if the settings have been maintained accordingly in transaction SMSY_SETUP). For systems not connected to the SLD, or for non-SAP products, the corresponding systems must be manually created in transaction SMSY in order to be able to use them in production operation in the solution landscape.



Systems that are unknown within the SLD can also be added manually using transaction SMSY.

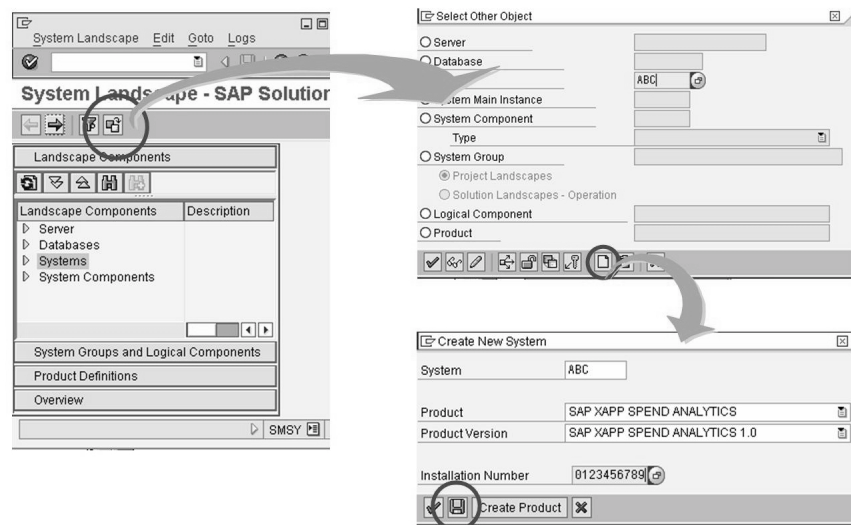


Figure 119: Addendum: Manually Creating Systems in Transaction SMSY

To do this, display the Solution Manager System Landscape (transaction SMSY). Then choose *Other Object...* in the *Landscape Components* area.

Result

As a result of the configuration steps introduced in this section, the system landscape (as it is defined in the SLD and, if necessary, supplemented by systems that have been added manually) is known to SAP Solution Manager. The system landscape defined in the SLD is periodically sent to the SAP Solution Manager system, which means that SAP Solution Manager is automatically informed about any changes to the systems landscape (for example, newly imported Support Packages).

Logical Components

In transaction SMSY (area *System Groups and Logical Components* → *Logical Components*), you need to assign the existing SAP systems to a logical component. A logical component is an administrative unit that assigns logical systems, in the entire system landscape and across projects, to the following elements:

- A main instance of a product with a product version, for example, the main instance *ECC Server* of the product *SAP ECC* with the product version *SAP ECC 6.0*.
- The system roles or phases in a project, for example, the system role development system for the configuration.



Caution: Note that the term “main instance” in SAP Solution Manager has nothing to do with the term “instance” in the technical sense (a bundle of processes that is started and stopped together). A main instance is a group of mutually dependent software component versions (from a technical perspective), installed and operated on a single server. Therefore, a main instance represents the smallest non-divisible unit within a system landscape.



A logical component assigns a logical system to the following throughout the system landscape:

- A “main instance” with a product version (for example, the “main instance” CRM Server of the product SAP CRM in product version 4.0)
- A system role or a project phase (for example, development system)

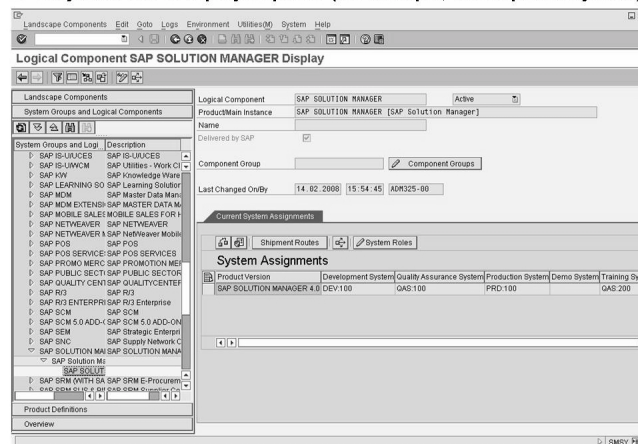


Figure 120: Logical Components and System Assignments

In this step, you can define, for example, which client in which SAP system is the development client, which client in which system is the quality assurance client, and whether or not there is a training client.

➡ **Note:** As a prerequisite for this task, the SAP systems must already be known to the SAP Solution Manager system.

RFC Destinations from the SAP Solution Manager System to the Connected Component Systems

Active RFC connections to the component systems are a prerequisite for the following Solution Manager functions:

- Generation of project IMGs in project management
- Navigation to the component systems in the configuration phase
- Tracking of change requests (change management)

➡ **Note:** In transaction SMSY, RFC connections can be generated for ABAP-based main instances only.

Depending on the scenario used, SAP Solution Manager must also have trusted RFC connections to the component systems. In this case, you must also assign trusted system authorizations to the relevant users. In particular, the authorization object *S_RFCACL* must be assigned.

The following figure shows how you can automatically create trusted RFC connections from the SAP Solution Manager system to the component systems.

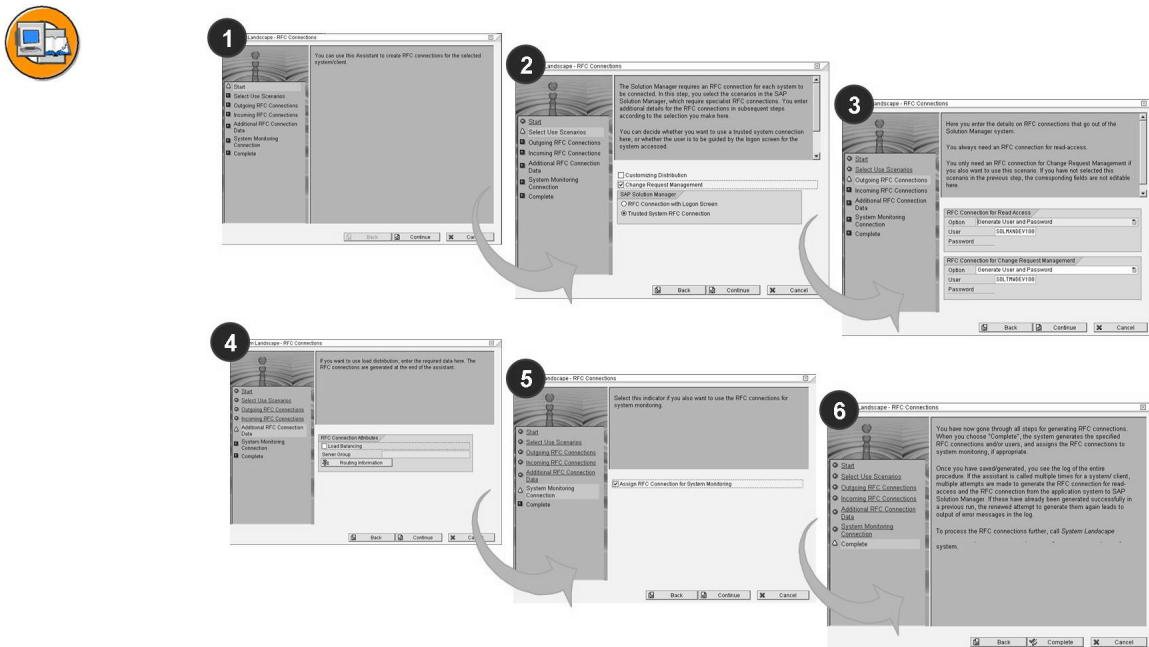


Figure 121: Creating RFC Connections to Component Systems

For this, call transaction SMSY and choose the area *Landscape Components* → *Systems* → <required SAP system type> → <SID> → <required SAP system type>. Switch to the *Clients* tab. In change mode, select the client to which you want to create the trusted RFC connection and choose *Generate RFC with Assistant*.

A wizard opens and guides you through the configuration process. At the end of this wizard, you are asked to log on several times to both the component system and the SAP Solution Manager system as a user who is allowed to create trusted RFC connections.

Result and Outlook

As a result of the configuration steps performed in this lesson, the system landscape (as it has been defined in the System Landscape Directory) has been made known to the SAP Solution Manager system. Furthermore, RFC connections have been established from the SAP Solution Manager system to the component systems that have been connected.

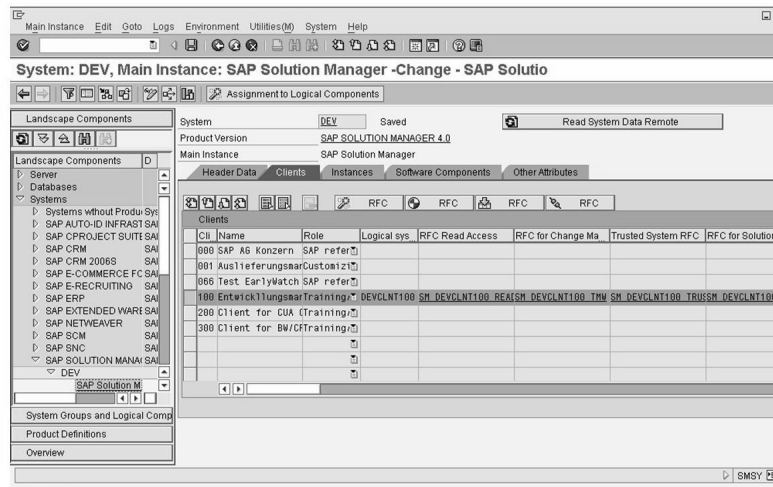


Figure 122: Result

The configuration settings of SAP Solution Manager (as they were introduced in this lesson) form, among other things, the basis for



- Monitoring systems and business processes in SAP Solution Manager
- Change request management with SAP Solution Manager

However, the configurations steps yet to be performed especially for these scenarios are beyond the scope of this lesson and, therefore, not discussed in detail here.

Examples of these additional configuration steps include:

- Creating an SAP Solution Manager project and generating IMG projects in the component systems for change request management (transaction SOLAR_PROJECT_ADMIN)
- Defining a solution in transaction SOLUTION_MANAGER or using transaction DSWP (a solution contains the systems in your system landscape that you have defined according to specific criteria, for example, all systems in production operation)

Exercise 13: Connecting ABAP-Based Systems to SAP Solution Manager

Exercise Objectives

After completing this exercise, you will be able to:

- Create a logical system and assign it to a client
- Connect ABAP-based SAP systems to the System Landscape Directory (SLD)
- Exchange data between the SLD and SAP Solution Manager
- Establish RFC connections from the SAP Solution Manager system to component systems

Business Example

Your company is currently implementing new SAP software. Various SAP systems have already been installed and configured as part of this implementation.

As a member of the SAP administration team, it is your responsibility to connect these SAP systems to the SAP Solution Manager system used centrally by your company. According to your planning, this will be done by connecting the relevant SAP systems to the System Landscape Directory (SLD) and then scheduling periodic data comparisons between the SLD and the SAP Solution Manager system.

Task 1: Creating a Logical System and Assigning It to A Client



The DEV and QAS systems play the role of both a satellite system (for example, an SAP ECC 6.0 system) and the central SAP Solution Manager system.

Team	Satellite Systems	SLD	SAP Solution Manager System
DEV	DEV and QAS	DEV	DEV
QAS	DEV and QAS	DEV	QAS

Figure 123: System Landscape in this Training Course

Create a logical system for your SAP system and assign this logical system to client 100 in your system.

1. In your SAP system, create the logical system `<SID>CLNT<client number>` where `<SID>` denotes your system ID and `<client number>` denotes the client in which you are working (for example, `DEVCLNT100` or `QASCLNT100`).
2. Assign the logical systems that you have created to the client that you are using in your SAP system for this course (for example, client 100).

Result

You have now created a logical system and assigned it to your client in your SAP system.

Task 2: Connecting an ABAP-Based SAP System to the System Landscape Directory

Configure the AS ABAP part of your SAP system in such a way that it periodically sends its system data to the System Landscape Directory (SLD). For this exercise, assume that the SLD is in the SAP system that has the system ID `DEV`. Use the gateway in the SLD system to transfer the data.

1. Use transaction RZ70 to check whether the connection from your SAP system to the SLD used in this course has been configured correctly. Activate this configuration and start the data collection and the job for transferring data to the SLD.

Continued on next page

2. Check whether the RFC destination *SLD_UC* exists and whether it refers to the SLD used in this course.
3. Log on to the graphical interface of the SLD used in your course and check whether the data in your SAP system has been transferred to the SLD.

Result

The list displayed should now contain your own SAP system, which has just sent its data to the SLD (*Last Update* column).

Task 3: For the DEV Groups Only: Creating Two Users in the System Landscape Directory

If, in your course, your system plays the role of the System Landscape Directory: In your SAP system, create a user *SLDUSER* who has administration authorization (UME role and security role *LCrAdministrator*) for the SLD. In addition, create the user **SAPJSF1** as a copy of the user **SAPJSF**.

1. Use transaction PFCG to check that your SAP system contains the PFCG role *SAP_SLD_CONFIGURATOR*. Generate an authorization profile for this role.
2. Use the UME console to check that the DEV system (in your SAP system) contains the UME group *SAP_SLD_CONFIGURATOR*, which corresponds to the PFCG role *SAP_SLD_CONFIGURATOR*.
3. In your SLD system (DEV system), use the Visual Administrator to assign the security roles *LCrInstanceWriterLD* and *LCrInstanceWriterNR* to the UME group *SAP_SLD_CONFIGURATOR* (component *sap.com/com.sap.lcr*sld*).
4. Use transaction SU01 to create the user **SLDUSER** as a *System User*. Assign the password **TRANSFER** and the role *SAP_SLD_CONFIGURATOR* to this user.
5. Create the user *SAPJSF1* as a copy of the user *SAPJSF* and assign the initial password **INIT01** to this user.

Result

You have now created the system user *SLDUSER* who has change authorization for the landscape definition and name reservation areas in the System Landscape Directory in your DEV system. Furthermore, you have created the user *SAPJSF1* as a copy of the user *SAPJSF*.

Continued on next page

Task 4: Transferring Data Between the System Landscape Directory and the SAP Solution Manager System

Configure the System Landscape Directory (SLD) and your SAP Solution Manager system (if necessary, using your partner group), so that you can schedule and execute periodic data transfers from the SLD to the SAP Solution Manager system.

1. Use transaction SMSY to check which systems are already known to the SAP Solution Manager.
2. In SAP Solution Manager, use transaction SLDAPICUST to maintain the SLD connection data. To do this, enter the user data of the user created in the previous task (**SLDUSER**).
3. **For the DEV group only:** In the *JCo RFC Provider* service within the Visual Administrator in the SLD system (DEV system), maintain an entry for the connection to the DEV system (<SID>=**DEV**) and an entry for the QAS system (<SID>=**QAS**). Maintain both entries as follows:

Field Name	Value
<i>Program ID</i>	SAPSLDAPI_<SID>
<i>Gateway host</i>	Your host name, for example, twdf9999
<i>Gateway service</i>	sapgw00
<i>Server count (1..20)</i>	1
<i>Application server host</i>	Your host name, for example, twdf9999
<i>System number</i>	00
<i>Client</i>	100
<i>Language</i>	EN
<i>User</i>	SAPJSF1
<i>Password</i>	INIT01

4. In your SAP Solution Manager system, use transaction SM59 to create the RFC destination **SAPSLDAPI** (type **T**). Here, define the program ID **SAPSLDAPI_<SID>** (as created in the SLD in the previous step) as a *Registered Server Program*. Perform a *Connection Test* for this RFC destination.

Continued on next page

5. Use transaction SMSY_SETUP to schedule data transfers from the SLD with *immediate* as the start time and *Daily* as the period. Use transaction SM37 to check whether the *LANDSCAPE FETCH* job has run successfully.
6. Use transaction SMSY again to check which systems are now known to SAP Solution Manager. Then save all of the systems in transaction SMSY again (area *Landscape Components* → *Systems* → *SAP SOLUTION MANAGER*).

Task 5: Logical Components

In your SAP Solution Manager system, use transaction SMSY to assign the client in which you are working to a suitable logical component and suitable role (development system or quality assurance system).

1. In the SAP Solution Manager system, use transaction SMSY to assign the logical component *SAP SOLUTION MANAGER* and the *Development System* role to client 100 in the DEV system.
2. In the SAP Solution Manager system, use transaction SMSY to assign the logical component *SAP SOLUTION MANAGER* and the *Quality Assurance System* role to client 100 in the QAS system.
3. Use transaction SMSY to check the current system assignments for the logical component *SAP SOLUTION MANAGER* in the *System Groups and Logical Components* area.

Task 6: RFC Destinations for the Component Systems

Use the wizard to create RFC connections from your SAP Solution Manager system to client 100 in your partner system (which, in this case, plays the role of a component system). If required, assign the necessary authorizations to the relevant SAP systems beforehand.

1. In client 100 in your SAP Solution Manager system, check that the role *Z_TRUST* exists and contains the necessary authorizations for creating trusted RFC destinations (authorization object *S_RFCACL*, choose *Utilities* → *Technical names on* to view the technical name of the authorization object). Assign this role to your user.
2. In client 100 in your partner system (the QAS system for the DEV groups and the DEV system for the QAS groups), check that the role *Z_TRUST* also exists here and contains the necessary authorizations for creating trusted RFC destinations (authorization object *S_RFCACL*). Assign this role to your user.

Continued on next page

3. In client 100 in your system, use transaction SMSY to create the RFC destinations to your partner system that are necessary for *Change Request Management*. Here, avail of the option to use a wizard to generate RFC destinations and specify your own SAP Solution Manager system as a *Trusted System*.
4. Optional: In client 100 in your SAP system, use transaction SMSY to create the RFC destinations to your own system that are necessary for *Change Request Management*. Here, avail of the option to use a wizard to generate RFC destinations and specify your own SAP Solution Manager system as a *Trusted System*.



Hint: The system may issue an error message because the required RFC connection back to the SAP Solution Manager system already exists from the previous substep.

Solution 13: Connecting ABAP-Based Systems to SAP Solution Manager

Task 1: Creating a Logical System and Assigning It to A Client



The DEV and QAS systems play the role of both a satellite system (for example, an SAP ECC 6.0 system) and the central SAP Solution Manager system.

Team	Satellite Systems	SLD	SAP Solution Manager System
DEV	DEV and QAS	DEV	DEV
QAS	DEV and QAS	DEV	QAS

Figure 124: System Landscape in this Training Course

Create a logical system for your SAP system and assign this logical system to client 100 in your system.

1. In your SAP system, create the logical system `<SID>CLNT<client number>` where `<SID>` denotes your system ID and `<client number>` denotes the client in which you are working (for example, **DEVCLNT100** or **QASCLNT100**).
 - a) If you have not yet done so, use the user assigned to you to log on to the client that you are using in the SAP system.
 - b) In the Implementation Guide (transaction SALE), choose *Basic Settings* → *Logical Systems* → *Define Logical System*, or start transaction BD54.



Hint: Alternatively, in the SAP Solution Manager system, you can display the *SAP Reference IMG* (transaction SPRO) and select the path *SAP Solution Manager* → *Configuration* → *Basic Settings* → *Standard Configuration of Basic Settings* → *Solution Manager* → *General Settings* → *Maintain Logical Systems*. Here, choose the *Define Logical Systems* substep.

- c) Choose *Edit* → *New Entries*.

Continued on next page

- d) In the *Log. System* column, create a new logical name (in upper case) for your logical system. For the logical system names, use the naming convention provided in the task description. Enter a description for the logical system (for example, **DEV system client 100**).
 - e) Save your entries, which will then be included in a transport request. If you are not assigned to a suitable transport request, create one now.
2. Assign the logical systems that you have created to the client that you are using in your SAP system for this course (for example, client 100).
- a) Call transaction SALE again. Choose *Basic Settings* → *Logical Systems* → *Assign Logical System to Client*, or call transaction SCC4 directly.
 - b) If necessary, switch to change mode. Select the row containing your client and choose *Details*.
 - c) Use the input help for the *Logical System* field to assign the logical system created earlier to your client. Save your entries.

Result

You have now created a logical system and assigned it to your client in your SAP system.

Continued on next page

Task 2: Connecting an ABAP-Based SAP System to the System Landscape Directory

Configure the AS ABAP part of your SAP system in such a way that it periodically sends its system data to the System Landscape Directory (SLD). For this exercise, assume that the SLD is in the SAP system that has the system ID *DEV*. Use the gateway in the SLD system to transfer the data.

1. Use transaction RZ70 to check whether the connection from your SAP system to the SLD used in this course has been configured correctly. Activate this configuration and start the data collection and the job for transferring data to the SLD.
 - a) In your SAP system, call transaction RZ70.
 - b) Check that the following settings have been selected, which permit the transfer of data to the SLD in the DEV system on your host:

<i>Transport Information</i>	<i>Automatic RFC Destination</i>
<i>Other Settings</i>	<i>Schedule background job with a frequency of 720 minutes.</i>
SLD Bridge: Gateway <i>Host</i>	<Your host name, for example, twdf9999 >
SLD Bridge: Gateway <i>Service</i>	sapgw00
Data Collection Programs	In the <i>Active</i> column, select all of the entries except <i>_SLD_RFC</i> .

- c) Choose *Activate Current Configuration* to activate your settings.
- d) Choose *Start Data Collection and Job Scheduling* to start the transfer of data to the SLD and confirm the next dialog box with *Yes*.
- e) Since the data transfer scheduled using *Start Data Collection and Job Scheduling* will not take place for another few hours, choose *Back*, *Schedule Job*, and then confirm the next dialog box with *Yes*. This schedules the job *SAP_SLD_DATA_COLLECT*, which will start in approximately one minute.

Continued on next page

2. Check whether the RFC destination *SLD_UC* exists and whether it refers to the SLD used in this course.
 - a) Call transaction SM59 and open the *TCP/IP connections* folder. Check whether the RFC destination *SLD_UC* exists with the following properties:

<i>Activation Type</i>	<i>Registered Server Program with the Program ID SLD_UC</i>
<i>Start Type of External Program</i>	<i>Default Gateway Value</i>
<i>CPI-C Timeout</i>	<i>Default Gateway Value</i>
<i>Gateway Host</i>	<Host name of your SAP system, for example, twdf9999>
<i>Gateway Service</i>	sapgw00



Hint: If this RFC destination did not already exist, it was automatically generated by the data transfer job started in the previous step.

3. Log on to the graphical interface of the SLD used in your course and check whether the data in your SAP system has been transferred to the SLD.
 - a) Open a Web Browser and navigate to the start page of the SLD in the DEV system on your host (<http://<your host name, for example, twdf9999.wdf.sap.corp>:50000/sld>). Here, 50000 denotes the http port for the AS Java part of the DEV system.
 - b) Log on to the SLD using the user and password assigned to you for the DEV system.
 - c) Navigate to the *Landscape* → *Technical Systems* area and select *Web AS ABAP* as the *Technical System Type*.

Result

The list displayed should now contain your own SAP system, which has just sent its data to the SLD (*Last Update* column).

Continued on next page

Task 3: For the DEV Groups Only: Creating Two Users in the System Landscape Directory

If, in your course, your system plays the role of the System Landscape Directory:
In your SAP system, create a user **SLDUSER** who has administration authorization (UME role and security role *LCrAdministrator*) for the SLD. In addition, create the user **SAPJSF1** as a copy of the user **SAPJSF**.

1. Use transaction PFCG to check that your SAP system contains the PFCG role **SAP_SLD_CONFIGURATOR**. Generate an authorization profile for this role.
 - a) In your SAP system, call transaction PFCG and use the input help to select the (single) role **SAP_SLD_CONFIGURATOR**. Then choose *Display*.
 - b) Switch to the *Authorizations* tab page. Here, choose *Display Authorization Data*. On the next screen, choose *Generate*.
 - c) Then choose *Back* and exit transaction PFCG.
2. Use the UME console to check that the DEV system (in your SAP system) contains the UME group **SAP_SLD_CONFIGURATOR**, which corresponds to the PFCG role **SAP_SLD_CONFIGURATOR**.
 - a) Use your Web Browser to start the UME console (**http://<your host name, for example, twdf9999.wdf.sap.corp>:50000/useradmin**) in your DEV system and then use your user to log on to the UME console.
 - b) On the initial screen of the UME console, start the search for *Groups* with the ID **SAP_SLD_CO***. The UME group **SAP_SLD_CONFIGURATOR** should be displayed as the search result.
 - c) Select the group **SAP_SLD_CONFIGURATOR**. The *Data Source* column should contain the entry **R3_ROLE_DS**, which shows that this UME group is also a PFCG role in AS ABAP.

Continued on next page

3. In your SLD system (DEV system), use the Visual Administrator to assign the security roles *LCrInstanceWriterLD* and *LCrInstanceWriterNR* to the UME group *SAP_SLD_CONFIGURATOR* (component *sap.com/com.sap.lcr*sld*).
 - a) Start the Visual Administrator in your SAP system at operating system level (*G:\usr\sap\DEV\DVEBMGS00\j2ee\admin\go.bat*).
 - b) If there is no connection to your DEV system, create one now by choosing *New* and then *Direct connection to a dispatcher node* (select **50004** as the port and use your user ID as the *User Name*). Choose *Save* to save your entry.
 - c) Select the connection that you have just created and choose *Connect* to log on to your DEV system.
 - d) Switch to the area *Cluster* → *DEV* → *<a server node>* → *Services* → *Security Provider*.
 - e) In the *Runtime* → *Policy Configurations* area, select the entry *sap.com/com.sap.lcr*sld* in the *Components* area.
 - f) Now choose the *Security Roles* tab page and switch to change mode. Select the security role *LCrInstanceWriterLD*.
 - g) In the *Mappings* → *Groups* area, choose *Add* and search for the group name **SAP_SLD_CO***. In the list that is now displayed, select the entry *SAP_SLD_CONFIGURATOR* and choose *OK*.



Hint: The security role *LCrInstanceWriterLD* contains the change authorization for the landscape definition area in the SLD.

- h) Now select the security role *LCrInstanceWriterNR*. In the *Mappings* → *Groups* area, choose *Add* and search for the group name **SAP_SLD_CO***. In the list that is now displayed, select the entry *SAP_SLD_CONFIGURATOR* and choose *OK* again.



Hint: The security role *LCrInstanceWriterNR* contains the change authorization for the name reservation area in the SLD.

Continued on next page

4. Use transaction SU01 to create the user **SLDUSER** as a *System User*. Assign the password **TRANSFER** and the role *SAP_SLD_CONFIGURATOR* to this user.
 - a) In your SAP system, call transaction SU01. In the User field, enter **SLDUSER**, and choose *Create*.
 - b) On the *Address* tab page, maintain a descriptive name.
 - c) On the *Logon data* tab page, select the *System* user type and assign the initial password **TRANSFER**.
 - d) On the *Roles* tab page, use the input help to assign the (single) role *SAP_SLD_CONFIGURATOR* to the user. Choose *Save* to complete the settings.
5. Create the user *SAPJSF1* as a copy of the user *SAPJSF* and assign the initial password **INIT01** to this user.
 - a) In your DEV system, call transaction SU01. In the User field, enter **SAPJSF**. Choose *Copy*.
 - b) In the next dialog box, enter the user name *SAPJSF1* in the *To* field, leave all other fields unchanged, and choose *Copy*.
 - c) On the *Logon data* tab page, assign the *Initial password* **INIT01**, and then choose *Save*.

Result

You have now created the system user *SLDUSER* who has change authorization for the landscape definition and name reservation areas in the System Landscape Directory in your DEV system. Furthermore, you have created the user *SAPJSF1* as a copy of the user *SAPJSF*.

Continued on next page

Task 4: Transferring Data Between the System Landscape Directory and the SAP Solution Manager System

Configure the System Landscape Directory (SLD) and your SAP Solution Manager system (if necessary, using your partner group), so that you can schedule and execute periodic data transfers from the SLD to the SAP Solution Manager system.

1. Use transaction SMSY to check which systems are already known to the SAP Solution Manager.
 - a) In your SAP Solution Manager system, call transaction SMSY.
 - b) Expand the area *Landscape Components* → *Systems* → *SAP SOLUTION MANAGER* and check that the list displayed does not yet contain your partner system (QAS system if you are in the DEV group or DEV system if you are in the QAS group). Ignore any entries from systems other than DEV or QAS.
2. In SAP Solution Manager, use transaction SLDAPICUST to maintain the SLD connection data. To do this, enter the user data of the user created in the previous task (**SLDUSER**).
 - a) In your SAP Solution Manager system, call transaction SLDAPICUST for maintaining the SLD access data, and switch to change mode.
 - b) If necessary, make the following changes to the row in which the *Prim.* field is selected: Change the *Host Name* to the name of your training host (for example, **twdf9999**) and the value for *Port* to **50000** (the HTML port of the Java dispatcher of the central instance in the SLD system, that is, the DEV system).
 - c) Furthermore, change the *User* to the user created in the SLD in the previous task (**SLDUSER**) and change the password to its password (**TRANSFER**). (Delete the existing password first before entering your new password.)
 - d) Finally, choose *Save* to save your entries.
3. **For the DEV group only:** In the *JCo RFC Provider* service within the Visual Administrator in the SLD system (DEV system), maintain an entry for the connection to the DEV system (<SID>=**DEV**) and an entry for the QAS system (<SID>=**QAS**). Maintain both entries as follows:

Field Name	Value
<i>Program ID</i>	SAPSLDAPI_<SID>
<i>Gateway host</i>	Your host name, for example, twdf9999

Continued on next page

Field Name	Value
<i>Gateway service</i>	sapgw00
<i>Server count (1..20)</i>	1
<i>Application server host</i>	Your host name, for example, twdf9999
<i>System number</i>	00
<i>Client</i>	100
<i>Language</i>	EN
<i>User</i>	SAPJSF1
<i>Password</i>	INIT01

- a) If you have not already done so, log on to your SAP system at operating system level using the user provided by your instructor.
- b) Start the Visual Administrator for the DEV system (*G:\usr\sap\DEV\DVEB-MGS00\j2ee\admin\go.bat*).
- c) If there is no connection to your DEV system, create one now by choosing *New* and then *Direct connection to a dispatcher node* (select **50004** as the port and use your user ID as the *User Name*). Choose *Save* to save your entry.
- d) Select the connection that you have just created and choose *Connect* to log on to your DEV system (that is, to the SLD system).
- e) Switch to the area *Cluster → DEV → Server → Services → JCo RFC Provider* and choose the *Runtime* tab page. On the *Bundles* tab page, choose the *Specific Application Server* option, and fill the fields for the connection to the DEV system in accordance with the table listed in the task description (<SID>=**DEV**). Finally, choose *Set* to save your entries.
- f) Repeat the previous substep for the connection to the QAS system (<SID>=**QAS**). For this connection, also use the *Gateway service* **sapgw00** in the DEV system.

Continued on next page

4. In your SAP Solution Manager system, use transaction SM59 to create the RFC destination **SAPSLDAPI** (type **T**). Here, define the program ID **SAPSLDAPI_<SID>** (as created in the SLD in the previous step) as a *Registered Server Program*. Perform a *Connection Test* for this RFC destination.
 - a) In your SAP Solution Manager System, call transaction SM59 and choose *Create*.
 - b) Create the *RFC Destination* **SAPSLDAPI** in the SAP Solution Manager system (transaction SM59). Choose **T** as the connection type. Enter a meaningful *Description* for this RFC destination and choose *Continue*.
 - c) On the *Technical Settings* tab page, choose the *Registered Server Program* activation type with the program ID **SAPSLDAPI_<SID>** (as created in the SLD in the previous substep, <SID> corresponds to your system ID). For *CPI-C Timeout*, choose the *Default Gateway Value* option, and fill the fields for the gateway (*Gateway host* and *Gateway service*) in accordance with the data contained in the table from the previous subtask. Choose *Save* to save your entries.
 - d) To test the setup, now perform a *Connection Test* for this RFC destination. If errors occur, check the settings both in the Visual Administrator in the SLD (see the previous subtask) and in transaction SM59 in your system. After correcting the errors, restart the *JCo RFC Provider* service in the Visual Administrator.

Continued on next page

5. Use transaction SMSY_SETUP to schedule data transfers from the SLD with *immediate* as the start time and *Daily* as the period. Use transaction SM37 to check whether the *LANDSCAPE FETCH* job has run successfully.
 - a) In your SAP Solution Manager system, call transaction SMSY and choose *Goto → Setup System Landscape Maintenance*. Alternatively, you can call transaction SMSY_SETUP directly.
 - b) On the next screen, select *System and Landscape Directory (SLD)* and choose *Schedule Data Transfer from TMS/SLD*. Confirm the next dialog box with *Yes*.
 - c) On the next screen, determine that the job should be executed with *immediate* as the start time, and then choose *Periodic job*. Choose *Period Values* and then select *Daily* as the period. Save the period values and start time values.
 - d) Use the input help to select an *Output Device* that already exists and choose *Continue*. Confirm the next dialog box, if necessary.
 - e) Now call transaction SM37. On the initial screen, select jobs with the job name **LAND***, by user name *****. Leave all other settings unchanged and choose *Execute*.
 - f) You should now see that the *LANDSCAPE FETCH* job is currently *Active* or it already has the status *Finished*. If necessary, choose *Refresh* until the job has finished. It should take your training system take less than one minute to execute the job.
6. Use transaction SMSY again to check which systems are now known to SAP Solution Manager. Then save all of the systems in transaction SMSY again (area *Landscape Components → Systems → SAP SOLUTION MANAGER*).
 - a) In your SAP Solution Manager system, call transaction SMSY.
 - b) Expand the area *Landscape Components → Systems → SAP SOLUTION MANAGER* and check that the list displayed now contains your partner system (QAS system if you are in the DEV group or DEV system if you are in the QAS group). Ignore any entries from systems other than DEV or QAS.
 - c) In the area *Landscape Components → Systems → SAP SOLUTION MANAGER*, select the entry *DEV* for the DEV system and choose *Save*. To do this, you may have to choose *Display ↔ Change* first in order to switch to change mode. Also repeat this substep for the entry *QAS* for the QAS system.

Continued on next page

Task 5: Logical Components

In your SAP Solution Manager system, use transaction SMSY to assign the client in which you are working to a suitable logical component and suitable role (development system or quality assurance system).

1. In the SAP Solution Manager system, use transaction SMSY to assign the logical component *SAP SOLUTION MANAGER* and the *Development System* role to client 100 in the DEV system.
 - a) In your SAP Solution Manager system, call transaction SMSY.
 - b) Expand the area *Landscape Components* → *Systems* → *SAP SOLUTION MANAGER* → *DEV* → *SAP Solution Manager*. Right-click the entry *SAP Solution Manager* for the DEV system and choose *Assignment to Logical Components*. This starts a wizard that guides you through the creation process.
 - c) On the initial screen of the wizard, choose *Continue*.
 - d) On the next screen (the step *Assign/Create Logical Components for System DEV*), use the input help to choose the *Logical Component* (*SAP SOLUTION MANAGER* → *SAP Solution Manager* → *SAP SOLUTION MANAGER*). Use the input helps to select the role and the client mentioned in the task description. Then choose *Continue* to progress to the next step.
 - e) On the next screen (the step *Overview of logical components for system DEV*), check your entries, and choose *Continue* → *Complete* to end the wizard.

Continued on next page

2. In the SAP Solution Manager system, use transaction SMSY to assign the logical component *SAP SOLUTION MANAGER* and the *Quality Assurance System* role to client 100 in the QAS system.
 - a) In transaction SMSY, expand the area *Landscape Components → Systems → SAP SOLUTION MANAGER → QAS → SAP Solution Manager*. Right-click the entry *SAP Solution Manager* for the QAS system and choose *Assignment to Logical Components*. Once again, this starts a wizard that guides you through the creation process.
 - b) On the initial screen of the wizard, choose *Continue*.
 - c) On the next screen (the step *Assign/Create Logical Components for System QAS*), use the input help to choose the *Logical Component (SAP SOLUTION MANAGER → SAP Solution Manager → SAP SOLUTION MANAGER)*. Use the input helps to select the role and the client mentioned in the task description. Then choose *Continue* to progress to the next step.
 - d) On the next screen (the step *Overview of logical components for system QAS*), check your entries, and choose *Continue → Complete* to end the wizard.
3. Use transaction SMSY to check the current system assignments for the logical component *SAP SOLUTION MANAGER* in the *System Groups and Logical Components* area.
 - a) In transaction SMSY, expand the area *System Groups and Logical Components → Logical Components → SAP SOLUTION MANAGER → SAP Solution Manager → SAP SOLUTION MANAGER*. Double-click the entry *SAP SOLUTION MANAGER*.
 - b) On the *Current System Assignments* tab page, you should now see that both a logical component for the *Development System (DEV:100)* and a logical component for the *Quality Assurance System (QAS:100)* have been maintained for the *Product Version SAP SOLUTION MANAGER 4.0*.

Continued on next page

Task 6: RFC Destinations for the Component Systems

Use the wizard to create RFC connections from your SAP Solution Manager system to client 100 in your partner system (which, in this case, plays the role of a component system). If required, assign the necessary authorizations to the relevant SAP systems beforehand.

1. In client 100 in your SAP Solution Manager system, check that the role *Z_TRUST* exists and contains the necessary authorizations for creating trusted RFC destinations (authorization object *S_RFCACL*, choose *Utilities* → *Technical names on* to view the technical name of the authorization object). Assign this role to your user.
 - a) In your SAP system, call transaction PFCG. In the *Role* field, enter **Z_TRUST**. Then choose *Display*.
 - b) Switch to the *Authorizations* tab page and choose *Display Authorization Data*. Expand the structure displayed and check that this role has all of the authorizations for the “authorization check for RFC users (for example, a trusted system)” (authorization object *S_RFCACL*, choose *Utilities* → *Technical names on* to view the technical name of the authorization object). Choose *Back*.
 - c) Now choose the *User* tab page and switch to change mode. Enter your user in the *User ID* column and choose *Save* to save your entries. Then choose *User comparison* → *Complete comparison*.
2. In client 100 in your partner system (the QAS system for the DEV groups and the DEV system for the QAS groups), check that the role *Z_TRUST* also exists here and contains the necessary authorizations for creating trusted RFC destinations (authorization object *S_RFCACL*). Assign this role to your user.
 - a) Log on to your partner system using a user provided by the instructor. In this SAP system, call transaction PFCG. In the *Role* field, enter **Z_TRUST**. Then choose *Display*.
 - b) Switch to the *Authorizations* tab page and choose *Display Authorization Data*. Expand the structure displayed and check that this role has all of the authorizations for the “authorization check for RFC users (for example, a trusted system)” (authorization object *S_RFCACL*). Choose *Back*.
 - c) Now choose the *User* tab page and switch to change mode. Enter your user in the *User ID* column and choose *Save* to save your entries. Now choose *User comparison* → *Complete comparison*.
 - d) Then log off from your partner system again.

Continued on next page

3. In client 100 in your system, use transaction SMSY to create the RFC destinations to your partner system that are necessary for *Change Request Management*. Here, avail of the option to use a wizard to generate RFC destinations and specify your own SAP Solution Manager system as a *Trusted System*.
 - a) In your SAP Solution Manager system, call transaction SMSY and navigate to the area *Landscape Components* → *Systems* → *SAP SOLUTION MANAGER* → <SID of your partner system> → *SAP Solution Manager*.
 - b) Now choose the *Clients* tab page and switch to change mode.
 - c) Select the row for client 100 and choose *Generate RFC with Assistant*. A wizard opens. Choose *Continue*.
 - d) In the *Select Use Scenarios* step, select the options *Change Request Management* and *Trusted System RFC Connection*. Leave all other entries unchanged and choose *Continue*.
 - e) Leave the settings in the steps *Outgoing RFC Connections*, *Incoming RFC Connections*, and *Additional Data for RFC Connection Data* unchanged and choose *Continue* to confirm these three steps.
 - f) Make sure that the *Assign RFC Connection for System Monitoring* option is **not** selected in the *System Monitoring Connection* step, and choose *Continue* → *Complete* to end the wizard.
 - g) At the end of the wizard, the SAP systems involved prompt you to log on (several times) to your partner system and to your own system (that is, to the component system and to the SAP Solution Manager system). Here, enter your user data (that is, the data of the user to whom you assigned the role *Z_TRUST* in the previous step) in each system (for example, **TADM10-##**).
 - h) Finally, on the *Display logs* screen, choose *Back*.

Continued on next page

4. Optional: In client 100 in your SAP system, use transaction *SMSY* to create the RFC destinations to your own system that are necessary for *Change Request Management*. Here, avail of the option to use a wizard to generate RFC destinations and specify your own SAP Solution Manager system as a *Trusted System*.



Hint: The system may issue an error message because the required RFC connection back to the SAP Solution Manager system already exists from the previous substep.

- a) In your SAP Solution Manager system, call transaction *SMSY* and navigate to the area *Landscape Components* → *Systems* → *SAP SOLUTION MANAGER* → *<SID of your own system>* → *SAP Solution Manager*.
- b) Now choose the *Clients* tab page and, if you have not already done so, switch to change mode.
- c) Select the row for client 100 and choose *Generate RFC with Assistant*. A wizard opens. Choose *Continue*.
- d) In the *Select Use Scenarios* step, select the options *Change Request Management* and *Trusted System RFC Connection*. Leave all other entries unchanged and choose *Continue*.
- e) Leave the settings in the steps *Outgoing RFC Connections*, *Incoming RFC Connections*, and *Additional Data for RFC Connection Data* unchanged and choose *Continue* to confirm these three steps.
- f) Make sure that the *Assign RFC Connection for System Monitoring* option is **not** selected in the *System Monitoring Connection* step, and choose *Continue* → *Complete* to end the wizard.
- g) At the end of the wizard, you are prompted to log on (several times) to your own system (in this case, your own system is both the SAP Solution Manager system and the component system). Here, enter your user data (that is, the data of the user to whom you assigned the role *Z_TRUST* in the previous step) in your system (for example, **TADM10-##**).
- h) Finally, on the *Display logs* screen, choose *Back*.



Lesson Summary

You should now be able to:

- Describe the role of the System Landscape Directory when mapping system landscapes
- List and explain the steps required to connect SAP Solution Manager to a System Landscape Directory
- Create logical components in SAP Solution Manager and assign systems
- Create RFC destinations in SAP Solution Manager for the component systems

Related Information

For more detailed information about configuring SAP Solution Manager, see:

- SAP Service Marketplace, quick link */instguides* in the area *Installation & Upgrade Guides* → *SAP Components* → *SAP Solution Manager* → *Release 4.0*, for example, in the document *Configuration Guide SAP Solution Manager as of SP<xx>*
- The *SAP Reference IMG* in the SAP Solution Manager system (transaction SPRO) in the area *SAP Solution Manager* → *Configuration* → *Basic Settings* → *Standard Configuration of Basic Settings* → *Solution Manager* → *System Landscape*. Here, short descriptions are also provided for each of the individual configuration steps.
- The online documentation for SAP Solution Manager in the area *SAP Solution Manager* → *Basic Settings* → *Solution Manager System Landscape*



Unit Summary

You should now be able to:

- Describe the added value of the SAP Solution Manager in an SAP system landscape
- Describe the role of the System Landscape Directory when mapping system landscapes
- List and explain the steps required to connect SAP Solution Manager to a System Landscape Directory
- Create logical components in SAP Solution Manager and assign systems
- Create RFC destinations in SAP Solution Manager for the component systems

Related Information

Additional information about SAP Solution Manager is available on SAP Service Marketplace under the quick link */solutionmanager* or in the online documentation for SAP Solution Manager.



Test Your Knowledge

1. Which of the following statements about connecting SAP systems to the System Landscape Directory (SLD) are correct?

Choose the correct answer(s).

- ☐ A Transaction RZ70 is used to connect SAP systems based on AS ABAP to the SLD.
- ☐ B The Visual Administrator (SLD Data Supplier Service) is used to connect SAP systems based on AS Java to the SLD.
- ☐ C Only transaction RZ70 is used to connect SAP systems based on AS ABAP + Java to the SLD. You do not require the Visual Administrator for this connection.
- ☐ D If you import a Support Package (SP) into an SAP system connected to the SLD, you must manually trigger another data transfer to the SLD, so that the SLD also receives information about the new Support Package.

2. Which steps are required (among others) to establish periodic data exchanges between the System Landscape Directory and the SAP Solution Manager system?

Choose the correct answer(s).

- ☐ A The JCo RFC Provider service must be configured correctly in the SLD.
- ☐ B A user who has administration authorization in the SLD must be defined in the SAP Solution Manager system.
- ☐ C Transaction SMSY_SETUP must be used to schedule the LANDSCAPE_FETCH job periodically in the SAP Solution Manager system.
- ☐ D An RFC destination (type T) for the SLD must be established in the SAP Solution Manager system.



Answers

1. Which of the following statements about connecting SAP systems to the System Landscape Directory (SLD) are correct?

Answer: A, B

To connect SAP systems based on AS ABAP + AS Java to the SLD, you must use transaction RZ70 for the AS ABAP part of the system and the Visual Administrator for the AS Java part of the system. You can schedule periodic data exchanges with the SLD, so that the SLD is automatically informed about any new Support Package levels imported into the system.

2. Which steps are required (among others) to establish periodic data exchanges between the System Landscape Directory and the SAP Solution Manager system?

Answer: A, C, D

A user who has administration authorization in the SLD does not have to be defined in the SAP Solution Manager system, but must exist in the SLD.

Unit 7

System Monitoring and Troubleshooting AS ABAP

Unit Overview

This unit is an introduction to system monitoring for SAP NetWeaver AS ABAP. You will first learn about the basics of the monitoring architecture and how to use the CCMS Alert Monitor. This unit also deal with creating your own monitors, including remote systems, and maintaining threshold values.

All of the content in this unit refers to the functions of ABAP-based SAP systems. You will find an introduction to monitoring Java-based SAP systems in the next unit. Other specialized training courses deal with this topic in more detail.



Unit Objectives

After completing this unit, you will be able to:

- Explain the concepts of the CCMS Alert Monitoring Infrastructure
- Use the CCMS Alert Monitor to monitor your system
- Configure the central monitoring of remote systems
- Design and create your own monitors
- Activate threshold values that are suitable for your system environment
- List various trace options
- Perform simple traces in the SAP system
- Develop procedures for structured troubleshooting

Unit Contents

Lesson: Monitoring Architecture	355
Exercise 14: System Monitoring.....	365
Lesson: Including Remote Systems.....	369
Exercise 15: Include Remote Systems	373
Lesson: Creating Your Own Monitors	377

Exercise 16: Create Your Own Monitors	381
Lesson: Properties Variants and Threshold Values	386
Exercise 17: Properties Variants of Monitors	393
Lesson: Trace Options	398
Exercise 18: Trace Options	407
Lesson: Troubleshooting Procedure	410

Lesson: Monitoring Architecture

Lesson Overview

This lesson provides an introduction to monitoring as a significant component of the Computing Center Management System (CCMS). Terms such as monitoring tree element (MTE), monitoring object, and monitoring attribute are discussed in detail.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the concepts of the CCMS Alert Monitoring Infrastructure
- Use the CCMS Alert Monitor to monitor your system

Business Example

You want to ensure good performance for the processing of business processes. You therefore regularly monitor the SAP systems, and take preventative action if required.

Fundamentals

Initial questions about monitoring:



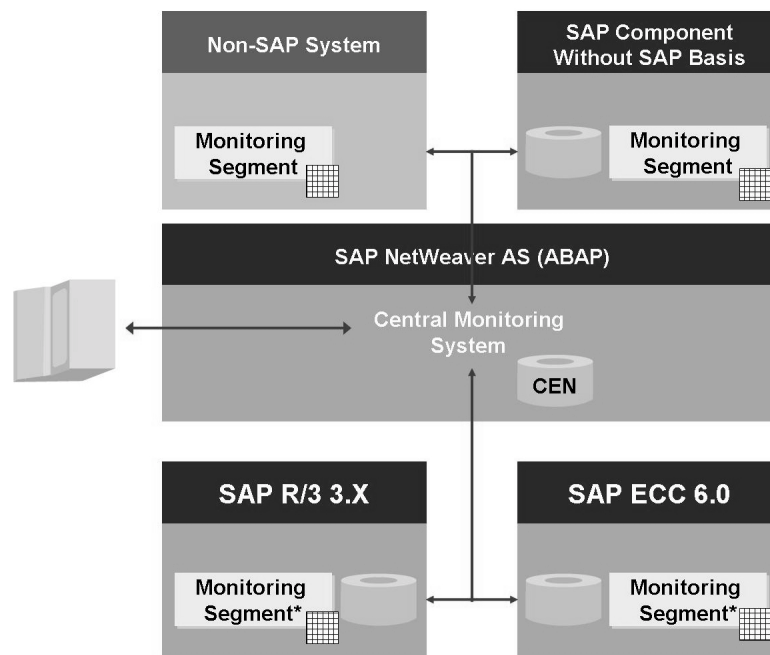
- Why?
 - To ensure the efficient processing of business processes
 - To ensure system security and stability
- How?
 - Central and cross-system
 - With an alert if an error occurs
 - With help that provides cross-system detailed information if an error occurs
- With which tool?
 - With the help of the CCMS Alert Monitoring Infrastructure and the special transactions connected to it

Nowadays, many components are usually involved in a business process. These components (whether produced by SAP or not) must be monitored, as both a gradual reduction in performance or a sudden breakdown of a component could affect overall productivity. It is a task of the administrator to monitor the system landscape regularly, and not only in the case of errors, but to take preventative action.

For example:

A file system where files of the SAP database are stored is 100% full. The database can no longer extend the tables in the files. A user performs a business transaction in the context of which a data record should be asynchronously added to one of these tables. The insert fails due to the space problem in the file system. The database error is seen as so serious that the entire asynchronous update process is automatically deactivated. All user sessions hang with the display of the hour glass. The SAP system hangs. If the fill level of the file system had been monitored regularly, the administrator could have taken action at the right time and system downtime could have been avoided.

Monitoring should be organized as efficiently as possible. There is not enough time for an administrator to log on to each host component to check its status. An efficient monitoring structure should be able to display the entire system landscape centrally at a glance. If an error occurs, the person responsible is automatically notified. Tools should be provided for the analysis of errors that provide cross-system detailed information about the problem.



* Within an SAP system, each instance has its own monitoring segment

Figure 125: Central Monitoring

The CCMS Alert Monitoring Infrastructure gives you the option of central and efficient monitoring for SAP systems.

The infrastructure must be installed on every component that is to be centrally monitored. This is automatically the case for SAP systems with software component SAP_BASIS 4.0 or above. SAP R/3 3.x systems and components on which no SAP system is active are connected using CCMS agents.

Each component collects its own monitoring data using the infrastructure and stores it locally in the main memory. This part of the main memory is called the **monitoring segment**. Its size can be configured.

One SAP system is selected as the central monitoring system. It should have as high a release level as possible and also be highly available. In large system landscapes, we recommend that you include a separate system that is used only for special tasks such as central monitoring, Central User Administration, transport domains controller, or the SAP Solution Manager. From a performance point of view, the workload of the central monitoring system increases only insignificantly, as the collection of monitoring data is usually decentralized.

The central monitoring system collects the monitoring data for the components and displays it in various views. In this way, the administrator has a central view of the entire system landscape. If errors occur, the administrator can jump directly from the central monitoring system (by RFC) to the relevant component to correct a problem in a detailed analysis.

Details

The CCMS Alert Monitoring Infrastructure consists of three parts: Data collection, data storage, and administration.

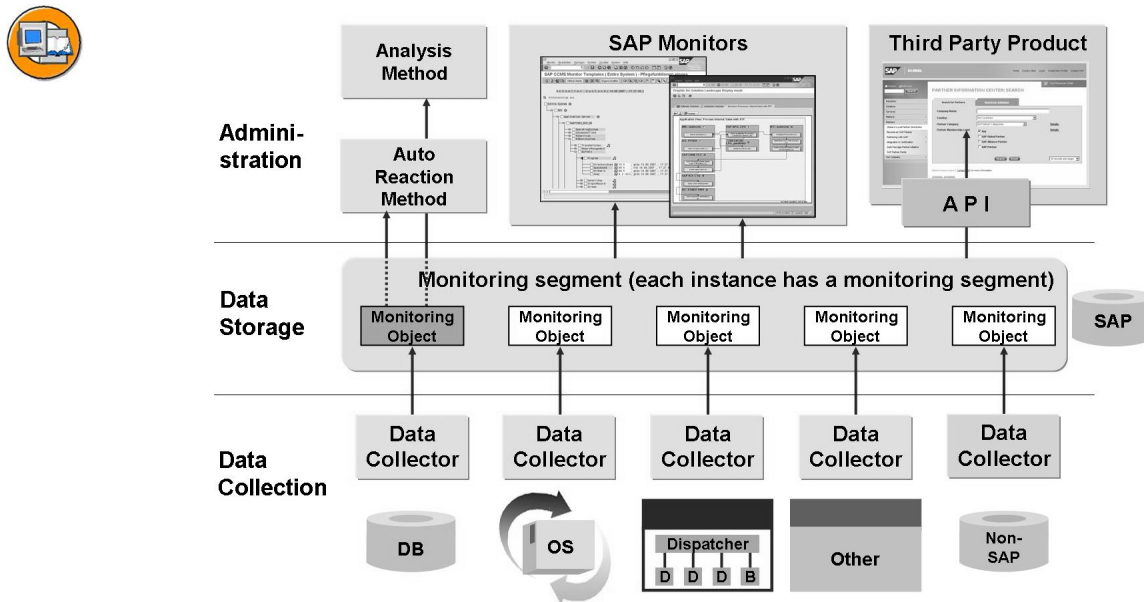


Figure 126: CCMS Alert Monitoring Infrastructure in Detail

At the **data collection** level, small subareas of an SAP system are monitored by special programs called data collectors. Data collectors can be ABAP, C, or Java programs. There are several hundred data collectors in ABAP alone. Each data collector checks its subcomponent at regular intervals and stores the collected monitoring data in the main memory of its host.

At the **data storage** level, the area of the main memory that contains the monitoring data from the data collector is called the monitoring segment. As the main memory data is always overwritten, it can be permanently copied to database tables. You can then analyze the data later. The data collection and storage elements must be present on every component that is to be centrally monitored.



Caution: Note that every instances of an SAP system has its own monitoring segment in shared memory. This means that for an SAP system with eight instances, there are eight different monitoring segments. The number of monitoring segments is determined by the number of instances. Whether or not several instances run on the same hardware, for instance, is of no significance here.

The **administration level** allows the data from the monitoring segment to be displayed and evaluated. SAP provides an expert tool, the CCMS Alert Monitor (transaction RZ20) as a display transaction. The SAP Solution Manager can show the data in a

business process-oriented context. If the system identifies a problem, it can execute a prepared automatic reaction, such as informing the responsible person. The analysis method then helps you to investigate the problem.

The CCMS Alert Monitoring Infrastructure can be extended. You can integrate your own components using data collectors that you have written yourself. Third-party vendors and partners can export the monitoring data from the monitoring segment using various interfaces.

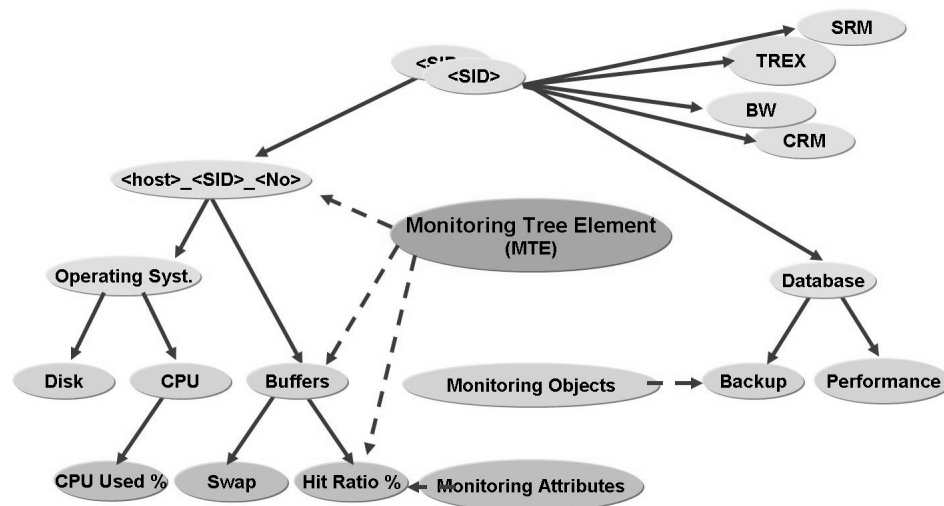


Figure 127: Monitor Structure

The CCMS Alert Monitor (transaction RZ20) displays the monitoring data from the monitoring segment in a tree structure. The tree structure allows a clear display when you are displaying a large number of measured values.

Any node in the tree is called a Monitoring Tree Element (MTE).

The measured values that are collected by the data collectors are displayed at the lowest level in the leaves of the tree. The leaves are known as monitoring attributes.

Threshold values can be stored for a monitoring attribute. SAP delivers default threshold values. However, in order to customize the monitor as well as possible for your system environment, you should check these threshold values, and adjust them if required.

Monitoring attributes are grouped at the second lowest level using monitoring objects. For example, the monitoring object *program buffer* contains, among others, the attributes *hit rate* and *swap*.

All other nodes in the tree serve to structure the monitoring objects in a logical and clear way, so that you can easily find the monitoring attribute that you require.

A CCMS monitor displays different subareas of the monitoring data. A monitor can contain data from multiple SAP systems.

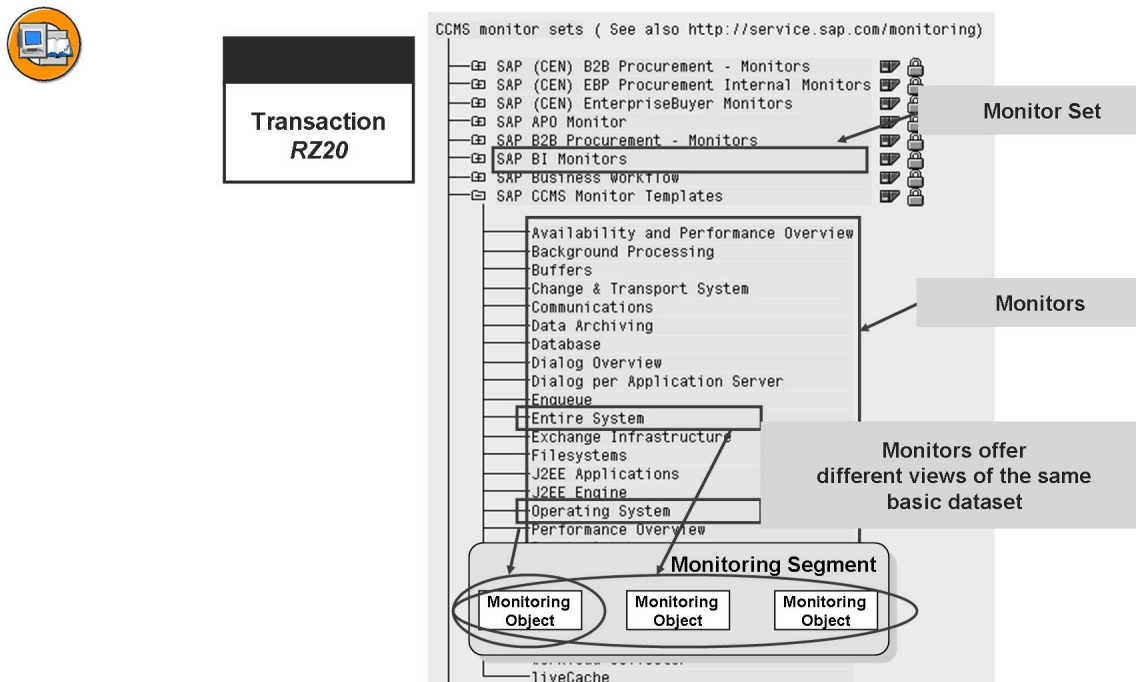


Figure 128: The CCMS Alert Monitor

You can access the monitor sets in the system by calling the transaction RZ20. Alternatively, in the SAP Easy Access menu, choose *Tools* → *CCMS* → *Control/Monitoring* → *CCMS Monitor Sets*.

SAP delivers preconfigured monitor sets that you can use immediately. Every monitor set bundles monitors that display various parts of the entire monitoring architecture, by topic area. It is therefore easier, for example, to find the database area.

The delivered monitor sets can be different for each system. For example, an SAP CRM system contains a special set for monitoring CRM scenarios. There are, of course, special data collectors connected with this. These are preconfigured and delivered with an SAP CRM system.

The monitoring data that monitors display can overlap. This means that the monitoring attribute *hit rate* of the program buffer can appear in several monitors. If you change, for example, the threshold value for this attribute in one of these monitors, it is changed in all monitors.

Some monitors, such as the monitor *Availability and Performance Overview* in the monitor set *SAP CCMS Monitor Templates*, do not display any data at first. This can be due to the fact that special settings are required to start the underlying data collectors.

To begin with, you will use the preconfigured monitors. Later, you can also create your own monitors that display exactly the data that you require for your daily monitoring work.

You can open a monitor by selecting its name.

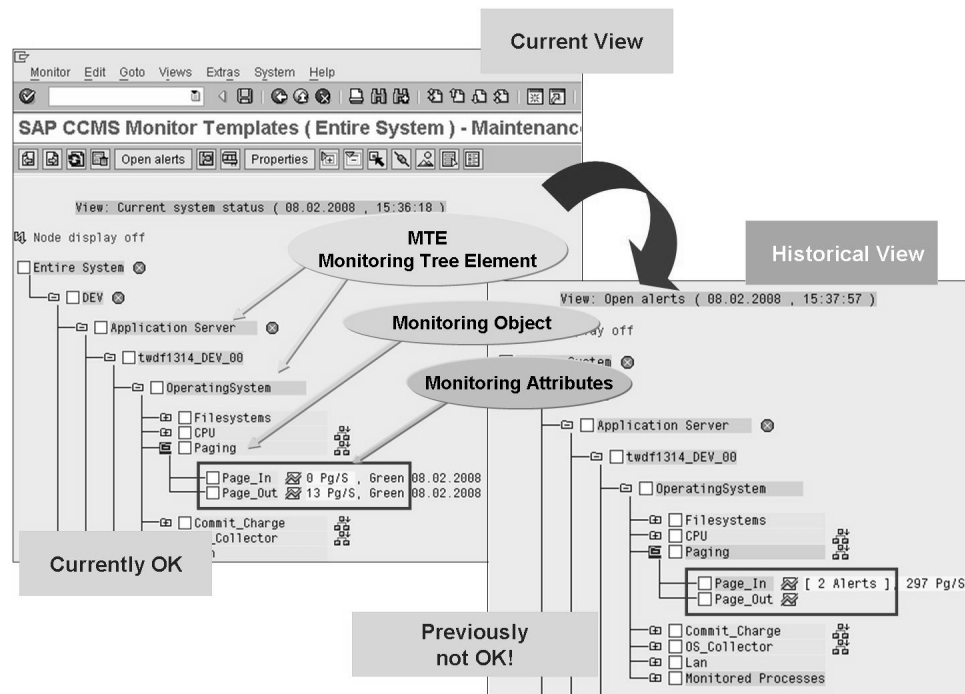


Figure 129: Layout of a Monitor

After you have opened a monitor, the corresponding monitoring data displays in the form of a tree. By clicking the “+” sign beside an MTE, you can expand the tree down to its leaves; the monitoring attributes.

Alert threshold values for triggering yellow and red alerts are assigned to monitoring attributes. If the threshold value condition is fulfilled, first a yellow, and then, if there is further deterioration, a red alert is triggered. The color of the monitoring attribute is propagated to its higher-level node in the tree, where the most severe alert is forwarded (red is more severe than yellow). This means that you can determine whether there is an alert in the tree from the root of the tree.

Views

The monitor should support you in your daily work. After you have opened the monitor, the following two views are available to you, amongst others:

- The *Current Status* view displays the monitor with the newest reported data.
- The *Open Alerts* view displays the monitor with its history information.

For example, during the previous night there may have been problems that are no longer occurring. In the *Current Status* view, the monitoring attribute is green, while it is displayed as red in the *Open Alerts* view. After you have ensured that there are currently no problems, you can then investigate problems that have previously occurred. You can see the selected view in the upper part of the monitor. You can switch views by choosing the *Current Status* or *Open Alerts* buttons.

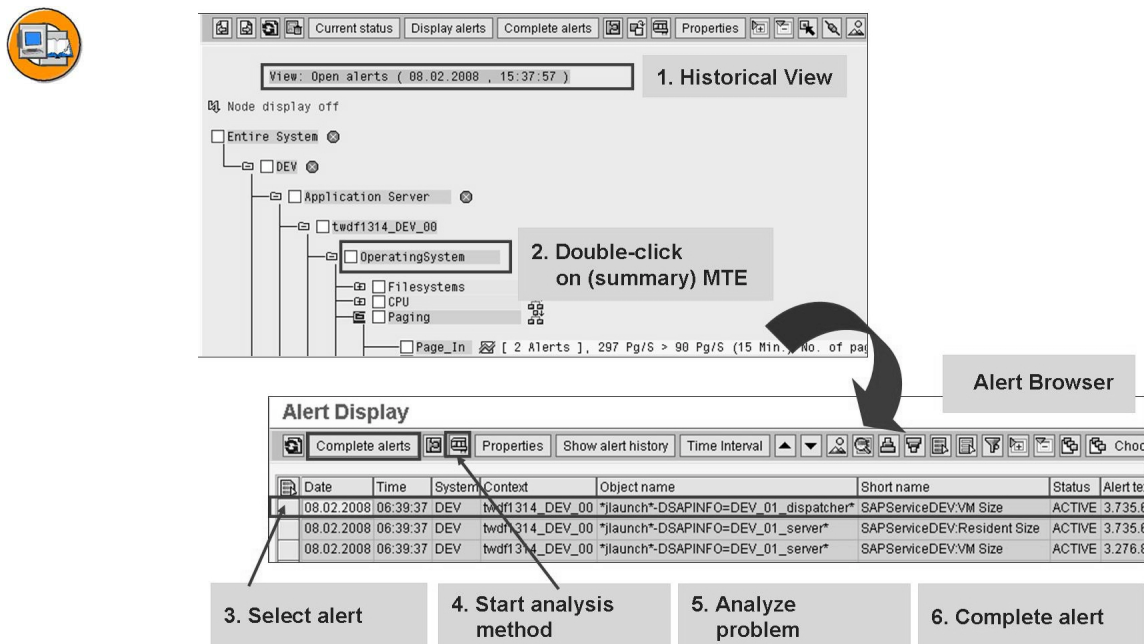


Figure 130: The Alert Browser

You can easily process the alerts that occurred in the past in the *Open Alerts* view. By double-clicking an MTE in the tree, you open the Alert Browser, which displays a list of all alerts for the selected MTEs and all alerts below it in the tree. This means that if you double-click the root of the tree, the system displays a list of all alerts in the tree, sorted by red and yellow alerts.

Select an alert that you want to process. Then choose the *Start Analysis Method* button. This starts the analysis method that is assigned to the MTE. The analysis method is a special tool that supports you when investigating problems. It can be transactions, or specially programmed function modules, or URL calls. You, therefore, do not need to remember all of the special tools, but simply use the CCMS Alert Monitor as a central point of entry.

After you have clarified the problem situation, choose *F3* to return to the Alert Browser. Then choose *Complete Alerts*. The processed alert is removed from the list and is stored in a database table.

Proceed in the same way with the remaining alerts, until the list is empty. When you next use your monitor, only the newly-occurred alerts display.

If you want to display completed alerts again, choose *Show Alert History* in the Alert Browser. Completed alerts display with the status *Done*.

Exercise 14: System Monitoring

Exercise Objectives

After completing this exercise, you will be able to:

- Evaluate and process alerts in the Alert Monitor

Business Example

You want to ensure good performance for business processes. You therefore regularly monitor the SAP systems, and take preventative action if required.

Task: The CCMS Alert Monitor

1. Start the CCMS Alert Monitor (transaction RZ20).
2. Open the *Entire System* monitor from the *SAP CCMS Monitor Templates* monitor set.
3. What is the current average dialog response time?
4. Switch to the *Open Alerts* view.
5. Select all alerts that have occurred in the *Dialog* area.
6. Process an alert from this list:

Start the analysis method for an alert.



Note: Analysis tools are not assigned to all attributes. If you know of any suitable analysis functions yourself, you can assign them to the attributes. This option is not dealt with as part of this course, however.

Return to the Alert Browser and complete the alert. Does the alert still appear in the list?

How can you display the completed alert again?

Solution 14: System Monitoring

Task: The CCMS Alert Monitor

1. Start the CCMS Alert Monitor (transaction RZ20).
 - a) Call the CCMS Alert Monitor (*Tools* → *CCMS* → *Control/Monitoring* → *CCMS Monitor Sets*, transaction RZ20).
2. Open the *Entire System* monitor from the *SAP CCMS Monitor Templates* monitor set.
 - a) Expand the *SAP CCMS Monitor Templates* set by choosing the “+” sign beside the set. Double click the *Entire System* set.
3. What is the current average dialog response time?
 - a) You can find the monitoring attribute for the average dialog response time by expanding, for example, the branch *<SID>* → *R/3 Services* → *Dialog* → *<Instance>*. Make sure that the monitor is in the *Current Status* view.
4. Switch to the *Open Alerts* view.
 - a) Choose *Open Alerts*.
5. Select all alerts that have occurred in the *Dialog* area.
 - a) Double click the *Dialog* MTE. All alerts in this area are displayed in the Alert Browser.
6. Process an alert from this list:
Start the analysis method for an alert.



Note: Analysis tools are not assigned to all attributes. If you know of any suitable analysis functions yourself, you can assign them to the attributes. This option is not dealt with as part of this course, however.

Return to the Alert Browser and complete the alert. Does the alert still appear in the list?

Continued on next page

How can you display the completed alert again?

- a) Select an alert in the list. Then choose the *Start Analysis Method* button. The system jumps from the monitor to a function that provides you with detailed data about the alert.

Return to the monitor. Choose *Complete Alerts*. The alert is removed from the list.

To display the alert again, choose Show Alert History. Your completed alert has the status *DONE*.



Hint: It is possible that the system displays other alerts with the status *AUTO_COMPLETE*. These alerts were completed automatically by the system to keep the alert area free for new alerts.



Lesson Summary

You should now be able to:

- Explain the concepts of the CCMS Alert Monitoring Infrastructure
- Use the CCMS Alert Monitor to monitor your system

Lesson: Including Remote Systems

Lesson Overview

In this lesson, you will learn how to include remote systems in the Alert Monitor.



Lesson Objectives

After completing this lesson, you will be able to:

- Configure the central monitoring of remote systems

Business Example

As an administrator, you want to include remote systems in the Alert Monitor.

Including Remote Systems

The monitors delivered by SAP display more detailed monitoring data for the local SAP system. Central system monitoring, on the other hand, has the advantage that you can monitor the entire system landscape at **one** glance, and not just your local system.

You can centrally monitor all components that have a CCMS monitoring infrastructure. SAP has delivered this infrastructure since SAP Basis 4.0. To include components that do not have an infrastructure, you can use the CCMS agent programs *SAPCM3X* for SAP R/3 3.x and *SAPCCMSR* for non-SAP components.

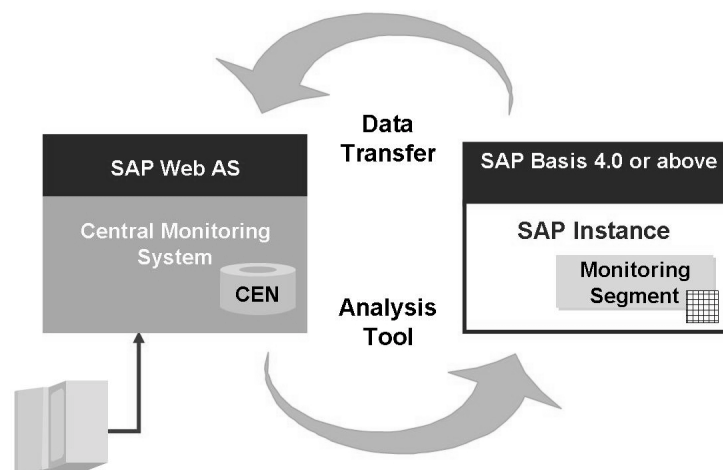


Figure 131: Including Remote Systems

To include an SAP system in a central monitoring architecture, you must define an RFC connection over which the monitoring data for the SAP system can be transferred to the central monitoring system. The data collection is performed independently by the CCMS monitoring infrastructure on the remote system.

From a security point of view, it is strongly recommended that you define a second RFC connection between the system with which the analysis tools can be started in the remote system from the central monitoring system. If a problem occurs, you can therefore branch directly from the central system to the remote system to analyze the situation in more detail.

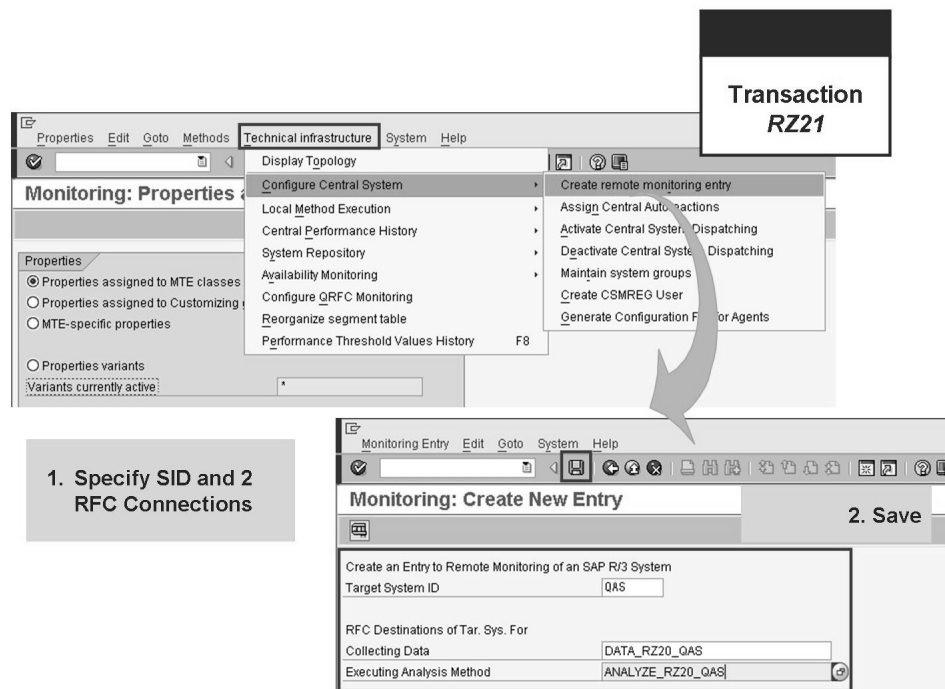


Figure 132: Including Remote Systems: Transaction RZ21

SAP systems are included in the central monitoring system in transaction RZ21. You can start the transaction from the SAP Easy Access menu by choosing *Tools* → *CCMS* → *Configuration* → *Attributes and Methods*.

In RZ21, choose *Technical Infrastructure* → *Configure Central System* → *Create Remote Monitoring Entry*.

In the *Target System ID* field, enter the SID of the SAP system to be monitored.

Now create the two RFC connections from the central monitoring system to the monitored SAP system. Choose *Goto* → *RFC Connections*. The system displays transaction SM59. Create two RFC connections here. Make sure that connection is of type “3”.

In the RFC connection for the transfer of the monitoring data, you can enter a user of the type *Communication* with a password that is valid in the monitored SAP system. This means that when the central monitoring system requests monitoring data from the monitored SAP system, it is provided without the need for user interaction. The user CSMREG is intended for this purpose.



Hint: You should create this user in the **remote system** in client **000** using a function in transaction RZ21. To do this, use the path *Technical Infrastructure* → *Configure Central System* → *Create CSMREG User*.

In the RFC connection that is used for the start of the analysis method, do not enter a user, but rather check the field *Current User*. If an analysis method is started in the monitored system from the central monitoring system when problems occur, callers must authenticate themselves in the monitored system.

Then choose *F3* to return to the transaction RZ21. Enter the RFC connections that you created under *RFC Destinations of Tar: Sys..* Choose *Save*. The SAP system can now be centrally monitored.



Hint: It makes sense to use the same naming convention for all the RFC connections you create for system monitoring. The following naming convention has proven effective for various reasons:

DATA_000_<SID> for the name of the RFC connection used to read data from the remote system

ANALYSIS_100_<SID> for the name of the “analysis RFC connection”. You should always use the RFC connection for reading data in the remote system with client 000. You can use any client for the analysis RFC connection. **<SID>** stands for the SID of the remote SAP system.

Exercise 15: Include Remote Systems

Exercise Objectives

After completing this exercise, you will be able to:

- Include a remote system in central system monitoring

Business Example

As a system administrator, you include remote systems in central system monitoring.

Task: Include Remote SAP Systems

Include remote SAP systems in the central monitoring architecture.

1. Your SAP system is the central monitoring system.



Hint: This exercise has two possible solutions.

On one hand, you can use the RFC connections `DATA_000_<SID>` and `ANALYSIS_100_<SID>` from the RFC unit, and on the other, you can follow the instructions for creating the RFC connections (if you have not performed the exercise in the RFC unit).

In the first case, you can continue directly with the entering of RFC connections in transaction RZ21.

Register the SAP system of your partner group in your system. To do this, get the details of a valid user with password from your partner group.



Hint: In any event you can use the user **Monitor** with the password **monitor**. It is created in clients 000 and 100.

Name the remote connections `DATA_000_<SID>` and `ANALYSIS_100_<SID>`.

Where can you see data from your partner group's SAP system?

Solution 15: Include Remote Systems

Task: Include Remote SAP Systems

Include remote SAP systems in the central monitoring architecture.

1. Your SAP system is the central monitoring system.



Hint: This exercise has two possible solutions.

On one hand, you can use the RFC connections `DATA_000_<SID>` and `ANALYSIS_100_<SID>` from the RFC unit, and on the other, you can follow the instructions for creating the RFC connections (if you have not performed the exercise in the RFC unit).

In the first case, you can continue directly with the entering of RFC connections in transaction RZ21.

Register the SAP system of your partner group in your system. To do this, get the details of a valid user with password from your partner group.



Hint: In any event you can use the user **Monitor** with the password **monitor**. It is created in clients 000 and 100.

Name the remote connections `DATA_000_<SID>` and `ANALYSIS_100_<SID>`.

Where can you see data from your partner group's SAP system?

- a) To do this, get the details of a valid user with password from your partner group.



Hint: In any event you can use the user **Monitor** with the password **monitor**. It is created in clients 000 and 100.

Call the configuration transaction for the CCMS Alert Monitor (*Tools* → *CCMS* → *Configuration* → *Attributes and Methods*, transaction RZ21).

Choose *Technical Infrastructure* → *Configure Central System* → *Create Remote Monitoring Entry*.

Under *Target System ID*, enter the `<SID>` of your partner group's system. Choose *Goto* → *RFC Connections*.

Continued on next page

Choose *Create*. Enter the following values:

- *RFC Destination*: **DATA_000_<SID>** (replace <SID> with the <SID> of your partner group's system)
- *Connection type*: **3**
- *Description*: Any documentation.
- Choose *Enter*.
- *Target host*: The host of your partner group.
- *System number*: The system number of your partner group.

On the *Logon/Security* tab page, enter the logon information that you received from your partner group for the user **CSMREG**. The password should be **monitor**.

Choose *Save*.

Choose *Test Connection*. If the connection test fails, check the data for *Target Host* and *System Number* with your partner group.

Choose *F3* to go back a step, and choose *Create* again.

Call the second RFC connection **ANALYSIS_100_<SID>** (replace <SID> with the <SID> of your partner group's system). Otherwise, enter the same data as for the RFC connection **DATA_000_<SID>**, with one exception: On the *Logon/Security* tab page, select *Current User* instead of entering the logon information.

Choose *Save*.

Perform a connection test again.

Return to transaction RZ21 by choosing *F3* twice.

In the *Data Collection* field, enter the RFC connection **DATA_000_<SID>**.

In the *Analysis Method* field, enter the RFC connection **ANALYSIS_100_<SID>**.

Choose *Save* and confirm the information dialog box that appears by choosing *Continue*.

A success message in the status bar tells you that the SAP system has been registered correctly.

You cannot yet see any data for the system of your partner group, as all monitors delivered by SAP display only local data.



Lesson Summary

You should now be able to:

- Configure the central monitoring of remote systems

Lesson: Creating Your Own Monitors

Lesson Overview

In this lesson, you will learn what to consider when designing and creating your own monitors.



Lesson Objectives

After completing this lesson, you will be able to:

- Design and create your own monitors

Business Example

As an administrator, you want to create a monitor that is specifically adjusted to your requirements.

Creating Your Own Monitors

Initial questions about designing your own monitors:



- Why should you build your own monitors?
 - To display exactly the values that are important for your daily work
 - To enable cross-system monitoring.
- How can you build your own monitors?
 - Consider what information you require
 - Create your own monitor set
 - Create some **static** monitors
 - Create some **rule-based** monitors (ADM106)
- Tips:
 - Design monitor definitions to deal with specific problems
 - Transfer as little data by RFC as possible
 - Note that you can transport monitor sets

SAP recommends that, for your regular work, you create your own monitors that display precisely the cross-system or local data that you require for your work.

The sets and monitors delivered by SAP cannot be changed. You should therefore first create your own monitor set. You can then create your own static monitors that display the required data.

The second technique for creating your own monitors, rule-based monitors, is described in detail in the follow-on course ADM106.

Before you create your own monitor, you should clarify the purpose of the monitor. The monitor should display as little data as possible in as clear a way as possible. You must make a selection that meets your requirements from the many hundreds of monitoring attributes that exist. A system overview monitor, for example, should contain the status of the last database backup or terminated updates as core indicators, but not details about the distribution of the dialog response time. You should create another monitor to display the response time.

Note also that the quantities of data to be transferred quickly become very large, especially if data from the monitored SAP systems is also to be displayed. A monitor that displays all monitoring data for multiple remote systems is unusable, as the data transfer can take too long, especially if the remote systems have a heavy load. As a global guide value, SAP recommends 10-20 monitoring attributes for each monitored instance in the central monitor.

Creating a Monitor Set

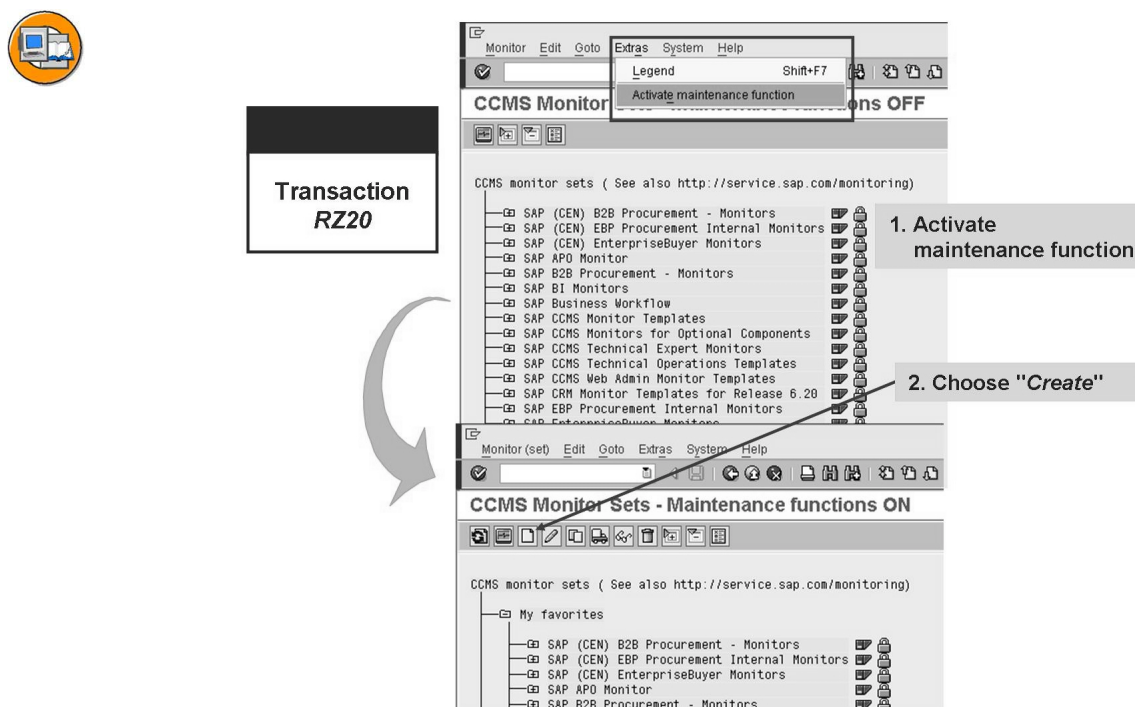


Figure 133: Creating a Monitor Set

The monitor sets (transaction RZ20) are usually in display mode, so you can open monitors, but cannot create or change them. To activate change mode, choose *Extras* → *Activate maintenance function* in transaction RZ20.

Maintenance functions ON appears in the transaction heading. The system displays new pushbuttons for creating and changing monitors and sets. Choose the *Create* button.

The system asks whether you want to create a monitor set or a monitor. Select *Monitor Set* and choose *Copy*. Enter a name for your monitor set.

Note the naming convention that your monitor set should not begin with SAP.

You can choose whether your set can be changed by other users or not. You can also choose whether the set is displayed directly for other employees in the monitor set overview.

After you have specified these attributes, choose *Enter*.

You have now created your own monitor set, into which you can either create new monitors or copy existing monitors.

The monitor sets and monitors delivered by SAP cannot be changed. However, you can use them as stable templates. You can set whether or not an SAP monitor set should appear in the display mode of the CCMS Alert Monitor.

If you want to hide an SAP monitor set, position the cursor (in *change mode*) on the name of the monitor set to be hidden (with a single click) and then choose *Change* (Shift + F1). Remove the selection for *public*. The set can now no longer be seen in the display mode. You can change your selection in the change mode.

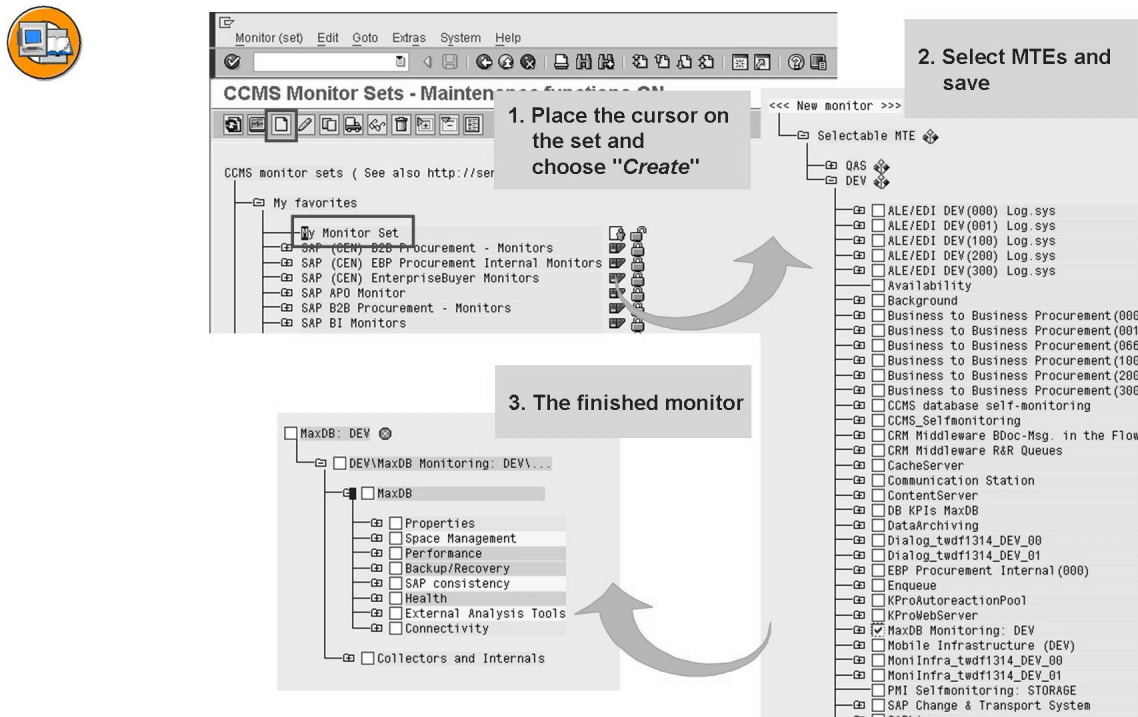


Figure 134: Static Monitors

Now create new monitors or copy existing monitors into your monitor set.

To create new monitors in your set, select the set and choose *Create*.

The system displays a selection screen in which all MTEs for all registered systems display.

Expand the tree structure and choose the MTEs that you want to display in your monitor by checking them. If an MTE is checked, all MTEs underneath it are automatically copied to the monitor. Take into account the number of MTEs.

Choose *Save*. The system prompts you for a name for the new monitor, which you can then start by selecting it.

You can organize your monitor more clearly by using virtual nodes when selecting the MTEs. Virtual nodes allow you to structure your monitor. During the MTE selection, choose *Create* and then *Virtual Node*. You can choose any text for the virtual node. It should be as descriptive as possible. Complete your entry by choosing *Enter*. Your virtual node is inserted. You can now select any MTEs for inclusion in the monitor under this node. In the final monitor, these MTEs appear under the virtual node.

Exercise 16: Create Your Own Monitors

Exercise Objectives

After completing this exercise, you will be able to:

- Create your own monitors

Business Example

As an administrator, you want to create a monitor that is specifically adjusted to your systems.

Task 1: Creating a Monitor Set

1. Create your own monitor set, System Monitoring.

Task 2: Create a Monitor

Create your own cross-system monitor, *Core_information* that displays the most important system monitoring data.

1. First consider what information is important for monitoring, from your point of view. Remember that you should not display too many values for each SAP system.
2. Now attempt to create the monitor *Core_information*. The monitor should, as far as possible, display the desired data for your system and your partner group's system.

Task 3: Process Alerts in the Remote System

1. Open your *Core_information* monitor. Display all alerts for the entire monitor.
Process an alert from your partner group's system.
 - a) Start the analysis method for an alert.
 - b) Return to the Alert Browser and complete the alert. Does the alert still appear in the list?
 - c) How can you display the completed alert again?

Result

You have just worked on a cross-system basis with the Alert Monitor.

Solution 16: Create Your Own Monitors

Task 1: Creating a Monitor Set

1. Create your own monitor set, System Monitoring.
 - a) Call the CCMS Alert Monitor (*Tools → CCMS → Control/Monitoring → CCMS Monitor Sets*, transaction RZ20).

Activate the maintenance function by choosing *Extras → Activate maintenance function*.

Choose the *Create* button. *New Monitor Set* is already selected. Choose *Copy*.

Enter the name **System Monitoring** for your monitor set. Maintain the attributes of your monitor set as you wish.

Choose *Copy*.

Your monitor set displays under *My Favorites*. You can now create new monitors in this set.

Task 2: Create a Monitor

Create your own cross-system monitor, Core_information that displays the most important system monitoring data.

1. First consider what information is important for monitoring, from your point of view. Remember that you should not display too many values for each SAP system.
 - a) Your new monitor, Core_information, should contain the most important monitoring data for your system landscape.

It could, for example, contain the following data:

 - Terminated updates
 - Terminated ABAP programs
 - System log messages
 - Average dialog response time

Continued on next page

2. Now attempt to create the monitor *Core_information*. The monitor should, as far as possible, display the desired data for your system and your partner group's system.

- a) In maintenance mode for transaction RZ20, place the cursor on your monitor set and choose *Create*.

The system displays the currently registered SAP systems.

Expand the tree and search for monitoring attributes that correspond to the data that you require.

For the proposed monitor from task 1:

- Terminated updates: Take a look in the branch
<SID> → <Instance> → R3Services → Update → Update → AbapErrorInUpdate.
- Terminated ABAP programs: Take a look in the branch
<SID> → <Instance> → R3Abap → Shortdumps.
- System log messages: Take a look in the branch
<SID> → <Instance> → R3Syslog
- Average dialog response time: Take a look in the branch
<SID> → <Instance> → R3Services → Dialog → ResponseTime.

Select the MTEs you require for both monitored SAP systems.

Choose *Save*. Enter **Core_information** as the name of your monitor.

Open your monitor.

Task 3: Process Alerts in the Remote System

1. Open your *Core_information* monitor. Display all alerts for the entire monitor.

Process an alert from your partner group's system.

- a) Start the analysis method for an alert.

- b) Return to the Alert Browser and complete the alert. Does the alert still appear in the list?

Continued on next page

c) How can you display the completed alert again?

a) Open your *Core_information* monitor .

Switch to the *Open Alerts* view.

Select an MTE for your partner system. All alerts in this area for your partner system display in the Alert Browser.

Select an alert in the list. Then choose the *Start Analysis Method* button. The logon screen of the partner system appears, as *Current User* is selected for the user data in the RFC connection for starting analysis methods.

Log on to your partner system. You can use the **Monitor** user with the password **monitor** for this again.

In the partner system, you jump to an action that provides you with detailed data about the alert.

Return to the monitor. Choose *Complete Alerts*. The alert is removed from the list.

To display the alert again, choose *Show Alert History*. Your completed alert has the status *DONE*.

Result

You have just worked on a cross-system basis with the Alert Monitor.



Lesson Summary

You should now be able to:

- Design and create your own monitors

Lesson: Properties Variants and Threshold Values

Lesson Overview

You can implement targeted monitoring of your systems using your own monitors that you create yourself. For these monitors to be able to work optimally, you must adjust the threshold values for the monitoring objects and attributes in the monitors for your system landscape. These specific settings can be defined as monitoring properties variants.



Lesson Objectives

After completing this lesson, you will be able to:

- Activate threshold values that are suitable for your system environment

Business Example

Monitors can only be meaningfully used if the selected threshold values for the individual monitoring attributes are set to sensible values. There are no generally recommended values, since different values make sense depending on the type of system, operation mode, usage type of the system (development/production), and your own expectations and requirements.

Threshold Values and Properties Variants

Introductory questions about properties variants and threshold values:



- Why?
 - So that alerts are not constantly or never triggered by the system
 - So that system monitoring is adapted in the best possible way to meet your requirements
- How?
 - In the central monitoring system in transaction RZ20
 - Transport from there to the monitored SAP systems
- Tips:
 - First create container for threshold values (properties variants) and activate
 - Then maintain threshold values for the MTEs in your own monitors

Threshold values can be stored for a monitoring attribute. Threshold values determine when the monitoring attribute should trigger a yellow or a red alert, and when it should become green or yellow again. The CCMS monitor infrastructure is delivered preconfigured with threshold values recommended by SAP. You should, however, check the threshold values, at least for the monitoring attributes that you consider to be important and that you have included in your own monitors. In this way, you adapt the system monitoring to your system environment in the best possible way. Otherwise, alerts can be constantly or never triggered, depending on whether the threshold value is too low or too high for your system environment.



Caution: Threshold values must be stored locally in every system.

However, instead of maintaining the same threshold values in every system, we recommend that you maintain the values in the central monitoring system and then distribute them to the monitored SAP systems using the transport system.

The prerequisite for transporting the threshold values to other SAP systems is that you have stored them in properties variants.



- What are properties variants?
 - Containers in which system monitoring settings can be saved. For example: Threshold values for an MTE
- What are properties variants used for?
 - Monitoring behavior can be switched dynamically
 - Monitoring behavior can be coupled to operation modes
 - Copying settings to other systems

You can create as many properties variants as you like and save settings in them. Just **one** properties variant with your settings is active at any time.

Properties variants have three advantages:

- You can manually switch from one properties variant to another for test purposes or to adjust the monitor to a special situation. This means that all monitor settings are automatically changed in accordance with the current properties variant.
- You can connect a properties variant to an operation mode. In this way, the threshold value (for a yellow alarm) for the dialog response time is set to 1400ms during the day, while the threshold value is automatically increased to 3500ms after the switch to night operation, since there is usually no dialog processing in your monitored system during the night.
- You can transport the contents of properties variants to other SAP systems using the transport system. For example, if you create a variant for production systems in the central monitoring system, and define the threshold values that are to apply for production systems there, you can then transport the variant to all production systems and activate the threshold values there.

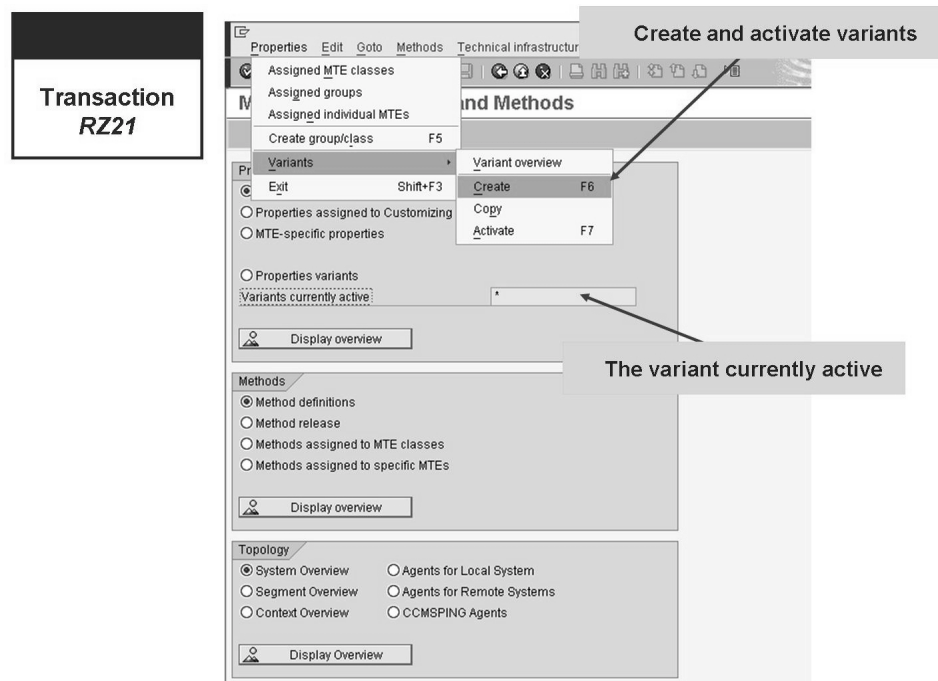


Figure 135: Creating and Activating Properties Variants

Properties variants are created in transaction RZ21. You can find the important functions for properties variants by choosing *Properties* → *Variants*.

First, create your own properties variant.

Choose *Properties* → *Variants* → *Create*.

Enter a name and a description for the properties variant, and save it.

You can organize properties variants hierarchically.

You can specify a *parent variant* when you create variants. If you do not specify another variant, the variant “*” is implicitly assumed to be the parent variant, whose parent variant, is, in turn, *SAP-DEFAULT*. The threshold values recommended by SAP are stored in *SAP-DEFAULT*.

Then choose *Properties → Variants → Activate*.

Select your variant and choose *Enter*.

Your properties variant that is now active, displays on the initial screen of transaction RZ21.

If no threshold value is maintained for a monitoring attribute in your variant, the system checks the parent variant. If this also has no threshold value, its parent variant is checked, and so on. Your properties variant is empty after it has been created. Therefore, after activation, the threshold values that are stored in the variant “*” or *SAP-DEFAULT* apply.

The connection of properties variants to operation modes is performed in transaction RZ04. Select the operation mode and choose *Operation Mode → Change*. You can enter the desired properties variant in *Monitoring Properties Variant*. Then save your entries.

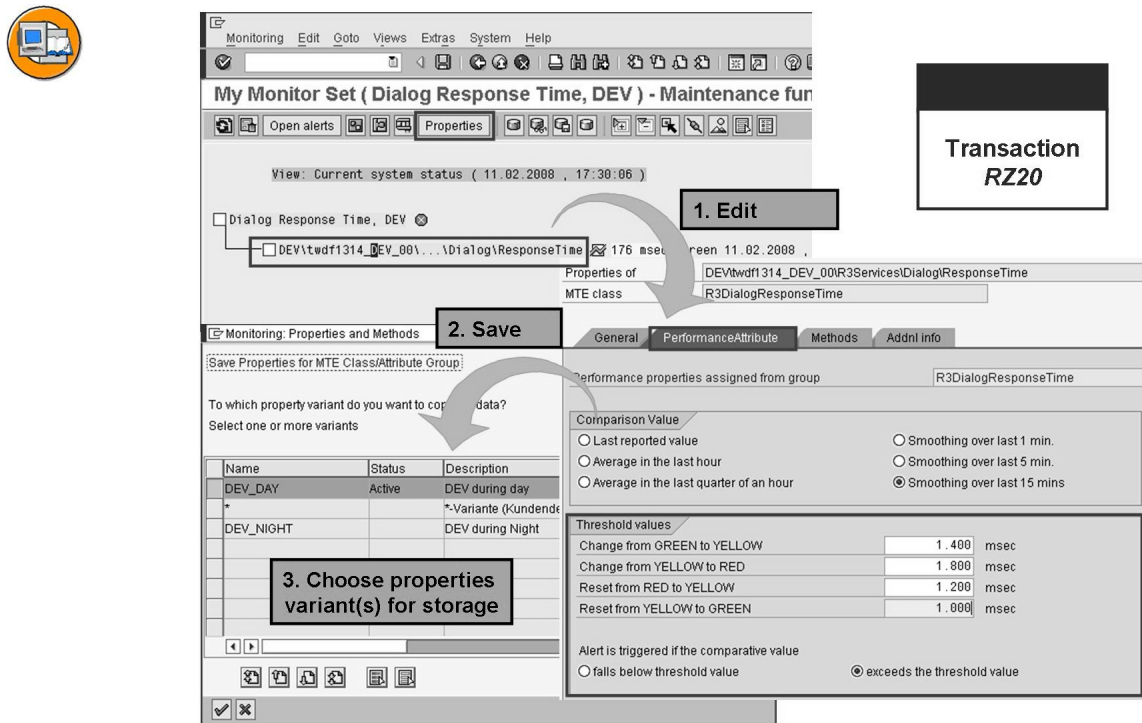


Figure 136: Maintain Threshold Values

After you have activated your properties variant, you can check the threshold values for the monitoring attributes you consider important and have included in your own monitors.

To do this, open your monitor(s). Select a monitoring attribute and choose *Properties*. The current tab page displays the valid threshold value definition.

The thresholds for *Change from GREEN to YELLOW* and *Change from YELLOW to GREEN* are defined to change sooner than the thresholds for *Change from RED to YELLOW* and *Change from YELLOW to GREEN*. In this way, you can avoid your monitor “flickering”, if the measured value is wavering around the threshold value. It is useful to give an **all clear** only once the situation has markedly improved.

Choose *Display* → *Change*. You can now adjust the threshold values to your requirements. Save your settings.

You can now choose, in a dialog box, in which properties variant your changed threshold values are to be stored. Note that the currently active variant is preselected. You can change the selection as desired.

You can copy these settings to a transport request and transport them to other SAP systems. To transport the threshold values, choose *Properties* → *Variants* → *Variant Overview* in transaction RZ21. In the variant overview, choose *Variant* → *Transport*. By doing this, you create a transport request that can be transported to other SAP systems using the transport management system (TMS).



Caution: Note that from a technical point of view, these transports can be made across releases and products. However, certain transport options have limited applicability. For example, you cannot import threshold values for attributes into an SAP system that for various reasons does not “recognize” these attributes (release, product). The import does not create inconsistencies in the target system, though.

You may need to process attributes in the various monitored systems locally (for example, threshold values for attributes that are only known in remote systems). It would be very helpful if you had already created the corresponding properties variants by transport for logging your changes.

Exercise 17: Properties Variants of Monitors

Exercise Objectives

After completing this exercise, you will be able to:

- Create a properties variant

Business Example

Monitors can only be used practically if the selected threshold values for the individual monitoring attributes are set to sensible values. There are no generally recommended values, since the requirements of SAP systems vary considerably and thus very different values may appear practical in different situations.

Task 1: Create a Properties Variant

1. Create your own properties variant, TEST, in your SAP system.
Activate your properties variant.

Task 2: Maintain Threshold Values

1. Maintain the threshold values for a monitoring attribute in your monitor. To do this, find a monitoring attribute from your SAP system, for which you can maintain the thresholds for:

Green to yellow

Green to red

Red to yellow

Yellow to green

Change these threshold values and save them.



Hint: These threshold values can only be defined for performance monitoring attributes. There are two other monitoring attribute types for which threshold values are specified in a different way.

Task 3: Switch Properties Variants Manually

1. Activate the “*” properties variant in transaction RZ21. What has happened to the threshold values for the MTE that you set in task 2?

Solution 17: Properties Variants of Monitors

Task 1: Create a Properties Variant

1. Create your own properties variant, TEST, in your SAP system.

Activate your properties variant.

- a) Call the configuration transaction for the CCMS Alert Monitor (*Tools → CCMS → Configuration → Attributes and Methods*, transaction RZ21).

Choose *Properties → Variants → Create*.

Enter **TEST** as the variant name and enter a short description.

Choose *Save*.

Choose *Properties → Variants → Activate*.

Select your variant *TEST* and choose *Enter*. The *TEST* variant is displayed as the active variant in the initial screen of transaction RZ21.

Task 2: Maintain Threshold Values

1. Maintain the threshold values for a monitoring attribute in your monitor. To do this, find a monitoring attribute from your SAP system, for which you can maintain the thresholds for:

Green to yellow

Green to red

Red to yellow

Yellow to green

Continued on next page

Change these threshold values and save them.



Hint: These threshold values can only be defined for performance monitoring attributes. There are two other monitoring attribute types for which threshold values are specified in a different way.

- a) To do this, call transaction RZ20 and select your monitor. Show a description of the symbols that are used in the monitor by choosing *Extras* → *Legend*. The icon for a performance attributes is a graph.

Search for a performance attribute for **your** SAP system in your monitor (such as the average dialog response time). Place the cursor on a monitoring attribute and choose *Properties*.

The current threshold values display on the *Performance Attribute* tab page. Do you consider the threshold values to be appropriate? Choose *Display <-> Change* and change the threshold values.

Choose *Save*.

In the *Monitoring: Properties and Methods* window, select your properties variant **TEST** and confirm by choosing *Continue*.

The threshold values are successfully adjusted and stored in your properties variant.

We recommend that you perform this check for all monitoring attributes for your system that are important for you. To activate the threshold values in other systems, you can transport your properties variant to these systems.

Continued on next page

Task 3: Switch Properties Variants Manually

1. Activate the “*” properties variant in transaction RZ21. What has happened to the threshold values for the MTE that you set in task 2?

- a) Call the configuration transaction for the CCMS Alert Monitor (*Tools* → *CCMS* → *Configuration* → *Attributes and Methods*, transaction RZ21).

Choose *Properties* → *Variants* → *Activate*.

Select the variant “*” and choose *Enter*. The “*” variant displays as the active variant in the initial screen of transaction RZ21.

Now open the *Core_information* monitor in transaction RZ20. Search for the monitoring attribute where you changed the threshold values in task 2.

The threshold values have returned to their original values, which are stored in the “*” properties variant.

If you reactivated your *TEST* properties variant in transaction RZ21, the changed settings would become active again.



Lesson Summary

You should now be able to:

- Activate threshold values that are suitable for your system environment

Related Information

You can find further information about system monitoring in:

- The SAP Service Marketplace: Quick Link */systemmanagement* → *System Monitoring and Alert Management*
- In the course ADM106 - *SAP System Monitoring Using CCMS I*. This course focuses, in particular, on the configuration of methods (analysis and auto-reaction) and the use of rule-based monitors.

Lesson: Trace Options

Lesson Overview

In this lesson, you will learn about the different trace options in the SAP system. You will perform and evaluate a trace yourself.



Lesson Objectives

After completing this lesson, you will be able to:

- List various trace options
- Perform simple traces in the SAP system

Business Example

An unexpected, reproducible error situation is occurring in your SAP system. As a member of the system administration team, it is your task to find the cause of the error.

Introduction

You can follow the process of various operations in your SAP system with trace functions. This allows you to monitor the system and isolate problems that occur.

There are many trace options in SAP systems. The main ones are listed below.



- System Log (SM21)
- Dump Analysis (ST22)
- System Trace (ST01)
- Performance Trace (ST05)
- Developer Traces (ST11)

You can use the **System Log** (transaction SM21) to detect and correct errors in your system and its environment. SAP application servers record events and problems in system logs. Every SAP application server has a local log that contains the messages output by this server.

If unpredictable errors occur during runtime when you call an ABAP program, a runtime error that generates a **short dump** can occur (transaction ST22).

If you want to record the internal SAP system activities, such as authorization checks, database accesses, kernel functions, and RFC calls, use the **System Trace** function (transaction: ST01).

The **Performance Trace** (transaction ST05) allows you to record database calls, lock management calls, and remote calls of reports and transactions in a trace file and to display the logged measurement results as lists. The Performance Trace also offers extensive support for a detailed analysis of individual trace records. You can find all the functions of the Performance Trace in the System Trace too. The Performance Trace is a more suitable analysis tool for certain problems, since the reduced scope of functions makes it easier to handle.

Technical information about internal SAP problems is logged in **developer traces**.

System Log

Events and problems are recorded locally on each application server and displayed in the system log in the SAP system.

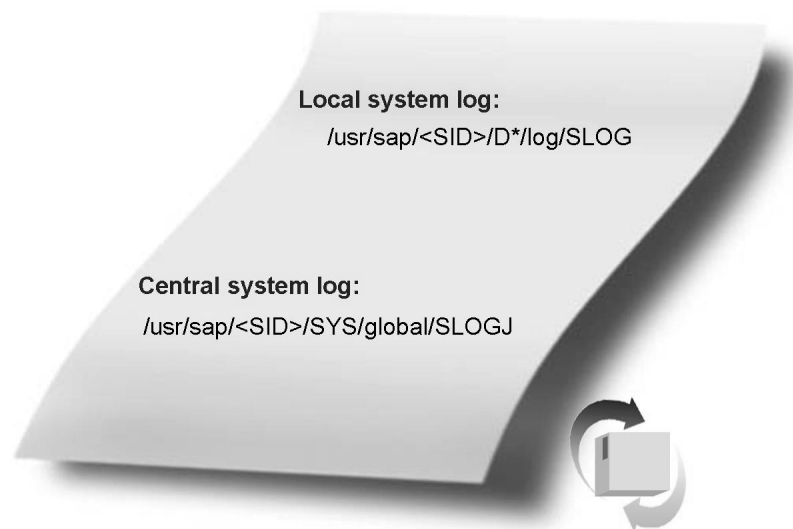


Figure 137: System Log (SM21)

If you are using the UNIX operating system, you can also work with central logging. In this case, each application server copies its local logs periodically to a central log. Central logging is not possible on Microsoft Windows and iSeries hosts. Technically, the system log is written to a ring buffer. If this log file reaches the maximum permitted size, the system begins to overwrite the oldest data.



Hint: The system does not display a message when an old log file is replaced.

To display a log, choose *Tools* → *Administration* → *Monitor* → *System Log* or call transaction SM21. By default, the system reads the log for the last one to two hours. As well as the local system log, you can display system logs for other application servers in transaction SM21. To do this, choose the menu path *System Log* → *Choose* → *All Remote System Logs* or *System Log* → *Choose* → *Central System Log*.

In expert mode (menu path *Edit* → *Expert Mode*), you can extend the selection criteria so that it is possible to search for entries for a particular terminal. To do this, choose the *Attributes* button.

In UNIX systems, you can display the status of the send process in the SAP system with transaction SM21 or by choosing *Environment* → *Process Status*.

You can define the path and file names for local and central log files with the following system profile parameters:

- *rslg/local/file*: File name for the local log (Default: SLOG<SAP_SYSTEM_NUMBER>)
- *rslg/central/file*: File name for the active central log (Default: SLOGJ); does not apply for Microsoft Windows NT and AS/400 platforms

By default, the log files for the local system log are stored in the following directory: */usr/sap/<SID>/<instance_directory>/log*. The central system log is stored in */usr/sap/<SID>/SYS/global*

You can also schedule system logging as a background job. There are two ABAP programs provided to do this:

- *RSLG0000*: To create the local system log
- *RSLG0001*: To create the central system log (not on Microsoft Windows NT and AS/400 platforms)

Dump Analysis

ABAP programs are checked statically when they are created and dynamically when they are running. Errors that are not statically predictable and only occur at runtime are dynamically identified by the ABAP runtime environment. States of this type lead to exceptions. If an exception is not handled or cannot be handled, a runtime error occurs. If a runtime error occurs, the ABAP runtime environment terminates the execution of the program, generates a short dump and branches to a special screen for analyzing the short dump. You can also find short dumps in transaction **ST22** or by choosing the menu path *Tools* → *ABAP Workbench* → *Test* → *Dump Analysis*.

A short dump is divided into different sections that document the error. The overview shows what other information is output in the short dump, such as contents of data objects, active calls, control structures, and so on. You can branch to the ABAP Debugger at the termination point from the short dump view. The following different error situations exist:

- **Internal Error**
The kernel identifies an error state. In this case, send a message to notify SAP.
- **Installation and Environment/Resource Error**
In this case, an error occurred that was caused by incorrect system installation or missing resources (such as the database being shutdown).
- **Error in Application Program**
Typical causes of errors are:
 - Content of a numerical field not in the correct format
 - Arithmetic overrun
 - An external procedure is not available
 - Type conflict when transferring parameters to an external procedure

By default, short dumps are stored in the system for 14 days. The transaction for managing short dumps is ST22. You can delete short dumps in accordance with a time specification using the *Reorganize* function, which you can call by choosing *Goto* → *Reorganize*. You can save a short dump without a time limit using the *Keep* function, which you can choose from the Detail View under *Short Dump* → *Keep/Release*.

If problems that you cannot solve yourself occur with ABAP programs, you can send an extract of the short dump to SAP. A short dump is the basis on which the SAP Hotline and remote consulting solve problems.

Important Characteristics of Dump Analysis



- If a runtime error occurs, a short dump is generated. You can use transaction ST22 to analyze this short dump.
- Dump data is stored in the database.
- Dump data can be reorganized.
- Individual short dumps can be flagged for retention.

(SAP) System Trace

You can use the (SAP) system trace (“system trace” for short) to record internal system activities. The system trace is primarily used if an authorization trace is to be created. SAP recommends that you use the system log or the developer trace for system monitoring and problem analysis. You can call the system trace using transaction ST01 or by choosing the menu path *Tools* → *Administration* → *Monitor* → *Traces* → *System Trace*. You can also use transaction ST01 to display the inactive trace file.

The system trace is used for analyzing:



- Authorization checks
- Kernel functions
- Kernel modules
- DB accesses (SQL Trace)
- Accesses to table buffers
- Lock operations (client-side)

You select the components to be logged on the initial screen. If the trace is activated for the authorization check, all authorization checks performed by the system are recorded. During the evaluation, you can identify which authorizations the system checked at which times. The following detail information is also provided: Date, time, work process number, user, authorization object, program, line, number of authorization values, authorization values.

You can use the SQL Trace to follow how the Open SQL commands in reports and transactions are converted to standard SQL commands and the parameters with which the SQL commands are transferred to the database system in use. The results of the SQL command are also logged, such as the return code and the number of records found, inserted, or deleted by the database. Logging the execution time and the callpoint in the application program allows you to perform more advanced evaluations.

With the enqueue trace, you can follow which lock instructions the SAP system performs on which lock objects, and which parameters the system uses for these locks. The program that triggered the lock, the owner of the lock, and the time that the enqueue server required to release the lock again are all also logged in the trace file.

You can use the RFC trace to follow which remote calls the SAP system executes, and the instance on which these calls are executed. You can see from the trace which function modules were called remotely by the program that is to be analyzed, and whether the RFC was successfully executed. The total time required for the execution of the remote call and the number of bytes sent and received during the RFC are also logged in the trace file.



Figure 138: System Trace (ST01) and Performance Trace (ST05)

Performance Trace

The Performance trace is used for analyzing:



- Database calls
- Lock management calls
- Accesses to table buffers
- Remote calls of reports and transactions
- Individual trace records
- SQL statements

The performance trace provides similar trace options to the system trace. It allows you to record database calls, calls to lock management, calls to table buffers, and remote calls of reports and transactions from the SAP system itself in a trace file. You can call the Performance Trace using transaction ST05 or by choosing the menu path *Tools → Administration → Monitor → Traces → Performance Trace*. On the initial screen of transaction ST05, you can choose the *Explain SQL* button to analyze an SQL statement without branching to a specific trace file.

The performance trace is integrated into the ABAP Workbench as a test tool and can therefore be called there.

Configuring the Trace File

You can use system profile parameters to restrict the size of the trace files and to specify an appropriate path.

The SAP system trace writes the trace data to trace files. For performance reasons, this is not done directly, but rather using a process-internal buffer. The profile parameter *rstr/buffer_size_kB* determines the size of the buffer. Since SAP Web AS 6.10, the SAP trace stores the data in multiple files, which are written in turn. The parameter *rstr/filename* defines the base name of these files. There is always a file with exactly this name. When the file becomes full (parameter *rstr/max_filesize_MB*), the file is renamed and a new file is generated with the base name. When the file is renamed, a number between 00 and 99 is added to the file name. The parameter *rstr/max_files* determines the maximum total number of files. If this value is exceeded, the files are overwritten.

Developer Trace

Developer traces are recordings that contain technical information and that are used if errors occur. This type of process trace is especially useful to investigate host and internal SAP problems that are affecting your SAP system. Developer traces **dev_*** are written to files in the directory */usr/sap/<SID>/<instance directory>/work* of the SAP application server that generated the trace.



Figure 139: Developer Traces

You can access the developer traces using the operating system, using transaction AL11, or from transaction SM50 (Work Process Overview). In transaction SM50, you can switch to the individual dev_* traces by choosing *Process* → *Trace* → *Display File*. You can display additional details in the displayed traces by expanding individual entries.

Exercise 18: Trace Options

Exercise Objectives

After completing this exercise, you will be able to:

- Use trace options in the SAP system to analyze the problem if errors occur
- Use the transactions for the various trace functions

Business Example

You want to use trace functions in the SAP system to correct errors.

Task: Traces

Activate traces in the SAP system and evaluate them. Generate a short dump in the system and analyze it.

1. In transaction ST01, activate the trace for authorization checks, RFC calls, and for lock operations for your user. Start the transaction for user maintenance SU01 and change the title for your own user.
2. Deactivate the trace again and evaluate the trace file.
3. In transaction ST05, activate the SQL trace for your user. Start transaction SA38 and execute the program **RSUSR000**. Deactivate the trace again and evaluate it.

Solution 18: Trace Options

Task: Traces

Activate traces in the SAP system and evaluate them. Generate a short dump in the system and analyze it.

1. In transaction ST01, activate the trace for authorization checks, RFC calls, and for lock operations for your user. Start the transaction for user maintenance SU01 and change the title for your own user.
 - a) Call transaction ST01 and select authorization check, RFC calls, and lock operations. In addition, you can choose the *General Filters* button to restrict the trace to your own user. Start the trace by choosing the *Trace on* button.
 - b) Call transaction SU01. Select your user and choose the menu entry *Users* → *Change*. On the *Address* tab page, change the title, and save your change.
2. Deactivate the trace again and evaluate the trace file.
 - a) Call transaction ST01 and choose the *Trace off* button.
 - b) To evaluate the generated trace file from transaction ST01, choose *Goto* → *Analysis* and select Authorization Check, RFC Calls, and Lock Operations. Start the analysis by choosing *Start Reporting*. You can now jump directly to the program code by selecting a data record and then choosing the *Go to ABAP Position* button.
3. In transaction ST05, activate the SQL trace for your user. Start transaction SA38 and execute the program **RSUSR000**. Deactivate the trace again and evaluate it.
 - a) Call transaction ST05 and select SQL Trace. Start the trace by choosing the *Activate Trace with Filter* button, and enter your user as the selection criterion.
 - b) Start the program **RSUSR000** in transaction SA38 and stop the trace in transaction ST05 by choosing *Deactivate Trace*.
 - c) To evaluate the generated trace file, call transaction ST05, choose the *Display Trace* button, and select SQL Trace. You can now display the SQL commands in plain text by selecting a data record with the operation “Open” or “Reopen” and choosing *Explain*.



Lesson Summary

You should now be able to:

- List various trace options
- Perform simple traces in the SAP system

Lesson: Troubleshooting Procedure

Lesson Overview

This lesson describes a general procedure for troubleshooting.



Lesson Objectives

After completing this lesson, you will be able to:

- Develop procedures for structured troubleshooting

Business Example

Unexpected problems are occurring during running operation of your SAP systems. As a member of the system administration team, you want to learn about the procedure for structured troubleshooting.

General Approach

It is part of their nature that errors always occur in places that they should not occur. It is therefore only possible to present a general approach. Front end printing is used to represent possible errors for examples.

You will run through the steps in the figure “Troubleshooting: Approach” when troubleshooting.

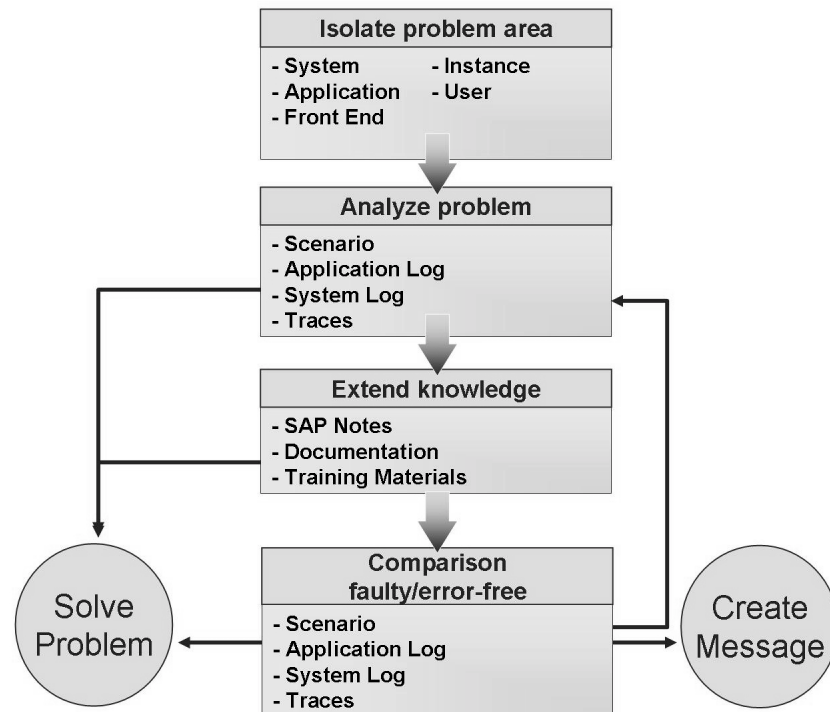


Figure 140: Troubleshooting: Approach

Isolate the problem area: First attempt to isolate the error. Where does it occur, when does it occur, and in what context does it occur? “It doesn't print”, would be too imprecise here. “Front end printing on front end xyz does not work with any SAP system”, is more exact. If you also know that front end printing works on other front ends, you have already isolated the problem.

Problem Analysis: Check the scenario to find out whether all required settings, and so on are correct. Check the application logs, the system log, and the traces (the developer traces will usually be helpful here), to see whether they provide any clues for correcting the error.

Gain additional knowledge: To interpret the results from the first problem analysis, it is, of course, necessary that you are familiar with the processes and functions of the area in which the error is occurring. If your experience and your previous knowledge are insufficient, you can start a search in the SAP Notes and on the SAP Service Marketplace with the keywords from the system log or the trace files. You may find a problem solution here, or additional information that helps you find and correct the error. If you have not found any suitable SAP Notes or suitable search terms, search for composite notes for the topic area. In this case, for example, with the terms *Front end printing* and *composite note*. For additional background information for the topic

area, see the online documentation and course materials. If you still cannot solve the problem with this information, use the comparison between the process with errors, and an error-free process.

Compare error-free and erroneous processes: You can use this to determine where there are differences between an erroneous and an error-free process. This information helps you to further isolate the problem area and may help you to solve the problem or to perform new, more targeted problem analyses. If it is not possible to perform another problem analysis, create a message for SAP on the SAP Service Marketplace. Enter the information from your troubleshooting (such as a trace and/or system log information) when doing so.



Lesson Summary

You should now be able to:

- Develop procedures for structured troubleshooting



Unit Summary

You should now be able to:

- Explain the concepts of the CCMS Alert Monitoring Infrastructure
- Use the CCMS Alert Monitor to monitor your system
- Configure the central monitoring of remote systems
- Design and create your own monitors
- Activate threshold values that are suitable for your system environment
- List various trace options
- Perform simple traces in the SAP system
- Develop procedures for structured troubleshooting



Test Your Knowledge

1. What can alert monitoring be used for?

Choose the correct answer(s).

- ☐ A Database backup
- ☐ B Updating data
- ☐ C Monitoring the database and the SAP system
- ☐ D Configuring and monitoring the firewall

2. Why do you incorporate remote systems in central system monitoring?

Choose the correct answer(s).

- ☐ A To transport program code from system to system
- ☐ B To create a local connection to a database backup of remote systems
- ☐ C To monitor these remote systems centrally
- ☐ D To allow file sharing
- ☐ E To connect an LDAP server

3. What types of monitors are there in the SAP system?

Choose the correct answer(s).

- ☐ A Ruled monitors
- ☐ B Statistical monitors
- ☐ C Rule-based monitors
- ☐ D Static monitors
- ☐ E Self-Repairing monitors

4. Properties variants are used to...

Choose the correct answer(s).

- ☐ A Store user master data
- ☐ B Customize transport requests
- ☐ C Save system monitoring settings (especially threshold values).
- ☐ D Store combinations of parameters for calling an ABAP report

5. With which of the following transactions can you activate a trace for SQL statements in the SAP system?

Choose the correct answer(s).

- ☐ A Performance trace
- ☐ B System log
- ☐ C (SAP) system trace
- ☐ D Database performance analysis



Answers

1. What can alert monitoring be used for?

Answer: C

Monitoring objects are only monitored, not administered using alert monitoring. The range of objects monitored is very large, since monitors can also be created for exotic objects.

2. Why do you incorporate remote systems in central system monitoring?

Answer: C

Including remote systems should allow you to monitor these systems centrally.

3. What types of monitors are there in the SAP system?

Answer: C, D

There are static and rule-based monitors.

4. Properties variants are used to...

Answer: C

Properties variants are used to save system monitoring settings, especially threshold values.

5. With which of the following transactions can you activate a trace for SQL statements in the SAP system?

Answer: A, C

You can analyze SQL statements by activating the trace in transaction ST01 (System Trace) or ST05 (Performance Trace). Transaction SM21 (System Log) is the system log and ST04 (Database Performance Analysis) is used to analyze database statistics.

Unit 8

Monitoring AS Java

Unit Overview

You can monitor SAP NetWeaver AS Java either locally in SAP NetWeaver AS Java itself or centrally using a central monitoring system (SAP NetWeaver AS ABAP). This unit shows both the local and central monitoring possibilities. It also provides a first insight into the areas of statistics and performance traces.



Unit Objectives

After completing this unit, you will be able to:

- List the SAP NetWeaver AS Java monitoring tools
- List the monitors that display data in a central system
- Describe the monitoring infrastructure
- Display monitoring data in the “Monitoring” service
- Make threshold value settings in the “Monitoring” service
- List the most important monitors in the Monitoring service
- Define which managers are involved in processing a request
- Monitor Java instances in the central monitoring system using an agent
- Install the SAPCCMSR agent for Java instances
- Explain which configuration steps are required to be able to maintain the threshold values for Java instances from the central monitoring system
- Operate the integrated and the central Log Viewer
- Explain the difference between logging and tracing
- Discuss the most important functions of the Log Configurator service
- Use the Log Configurator service to adjust the severity of log files
- Describe how an availability check using the GRMG works technically
- Configure an availability check
- Explain which steps a developer must perform to create a GRMG-compatible application

- List the different trace options
- List the different statistics options
- Discuss how traces are activated and where they are displayed
- Display the most important (Java) statistics in SAP NetWeaver AS ABAP
- List the functions of the SAP Solution Manager Diagnostics
- Understand the the architecture of the SAP Solution Manager Diagnostics

Unit Contents

Lesson: Java Monitoring: Overview	421
Lesson: Monitoring SAP NetWeaver AS Java	429
Exercise 19: Monitoring SAP NetWeaver AS Java	441
Lesson: Appendix: Background Information About the Monitoring Service	447
Lesson: Connecting to a Central Monitoring System	458
Exercise 20: Connecting to a Central Monitoring System	469
Lesson: Log Viewer and Log Configuration	473
Exercise 21: Log Viewer and Log Configuration	495
Lesson: Availability Monitoring	505
Exercise 22: Availability Monitoring	515
Lesson: Appendix: Statistics and the Performance Trace	519
Exercise 23: Statistics and the Performance Trace	537
Lesson: Appendix: Solution Manager Diagnostics (SMD)	541

Lesson: Java Monitoring: Overview

Lesson Overview

You can use various tools to monitor a SAP NetWeaver AS Java. You can display most collected monitoring data both locally in the SAP NetWeaver AS Java and in a central monitoring system. This lesson provides an overview of the available monitors. The subsequent lessons provide more detailed information.



Lesson Objectives

After completing this lesson, you will be able to:

- List the SAP NetWeaver AS Java monitoring tools
- List the monitors that display data in a central system

Business Example

You are using SAP NetWeaver AS Java and want to react quickly if errors occur. For this reason, you have decided to set up monitoring with SAP resources. Both local and central monitoring tools are available to you.

Overview of the Monitoring Options of SAP NetWeaver AS Java

Together with SAP CCMS, SAP NetWeaver AS Java provides a monitoring architecture that accumulates data, creates histories, and generates alerts. Some data can both be displayed locally and transferred to a central monitoring system using the SAPCCMSR agent.

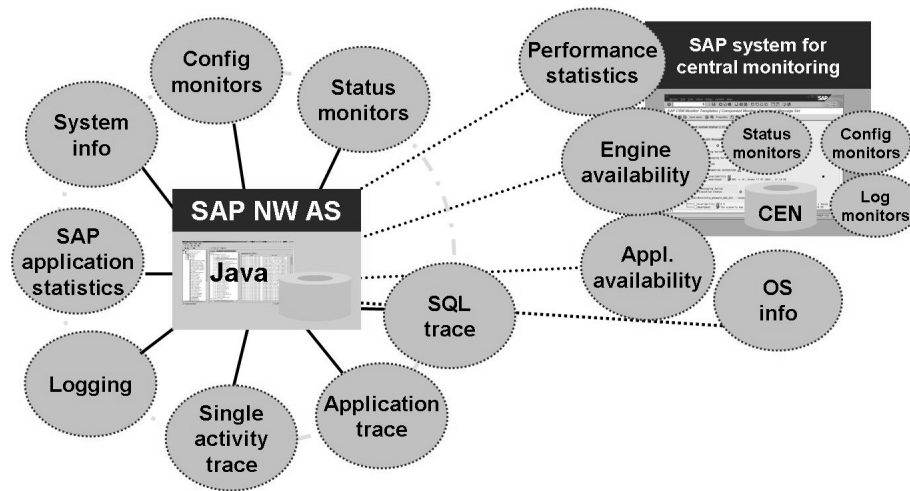


Figure 141: Monitoring Options of SAP NetWeaver AS Java

Local Monitoring of SAP NetWeaver AS Java

Each Java instance can access its own locally collected monitoring data. This is displayed in the infrastructure for monitoring and management of SAP NetWeaver AS Java. The infrastructure consists of several tools and incorporates the following functions:

- Monitoring service
- Logging using the Log Viewer
- Application Trace
- Single Activity Trace
- System Info
- SAP Application Statistics

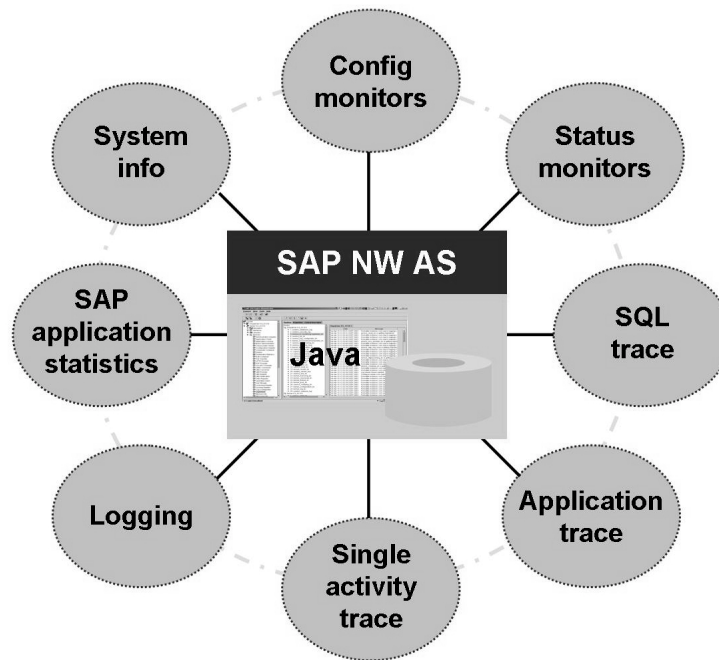


Figure 142: “Local” monitoring of the SAP NetWeaver AS Java

Monitoring service

The monitors created by the Monitoring Service of SAP NetWeaver AS Java monitor all critical server parameters. The Monitoring Service consists of status monitors and configuration monitors. The monitoring architecture used in the Monitoring Service is based on the JMX standard. This means that external tools can also access the monitoring data.

The most important resources of the SAP NetWeaver AS Java that are monitored are:

- Memory Use
- Threads
- All services and managers
- Database connections
- Database transactions
- Sessions

You can display the data collected from the monitored resources in the Monitoring Service of the dispatcher and the Monitoring Service of the servers in the Visual Administrator.

Logging

All important events that occur in a cluster node of SAP NetWeaver AS Java are written to log files. All Java components of SAP NetWeaver AS use the same infrastructure for logging. You can configure the logging at a shared location using the Log Configurator Service and display the logging using a shared tool - *the Log Viewer*. The Log Viewer allows you to quickly and efficiently find log files containing specific error and event information across multiple servers.

Application Trace

The Application Trace service is a tool for developers for debugging J2EE applications at runtime. To activate the trace, you need to restart the application. When you do so, the bytecode is changed so that certain markers are set that measure the time used by the individual Java methods. The Application Trace is integrated into the Performance Tracing service of the Visual Administrator as a service.

Single Activity Trace (SAT)

You can use the Single Activity Trace (SAT) to trace individual user requests that run distributed across multiple components. The collected data is based on *Java Application Response Measurement (JARM)*. You can activate the Single Activity Trace in the Performance Tracing service of the Visual Administrator, and display the Single Activity Trace using the Log Viewer.

SAP Application Statistics

If there are performance problems with an application, you can include each individual (user) request in the analysis. In this case, the collected data is based on *Java Application Response Measurement (JARM)*. This method collects the response times of a Java application. Information about the user that created the request and the quantity of data transferred is also available. The JARM data is displayed in the Visual Administrator in the *Performance Tracing* service.

SQL Trace

You can activate the SQL Trace dynamically for specific SQL statement to obtain an analysis of the SQL statements used. The SQL Trace is used for performance analysis, primarily during the development process. In addition to the SQL statement text, it provides information about the time, duration, result, and input parameters of the executed statement.

System Info

You can obtain an initial overview of the status of the Java cluster using the *System Info* application. With this application, you can see the status of the dispatchers and servers of local Java instances and obtain information about the release and Support Package statuses.

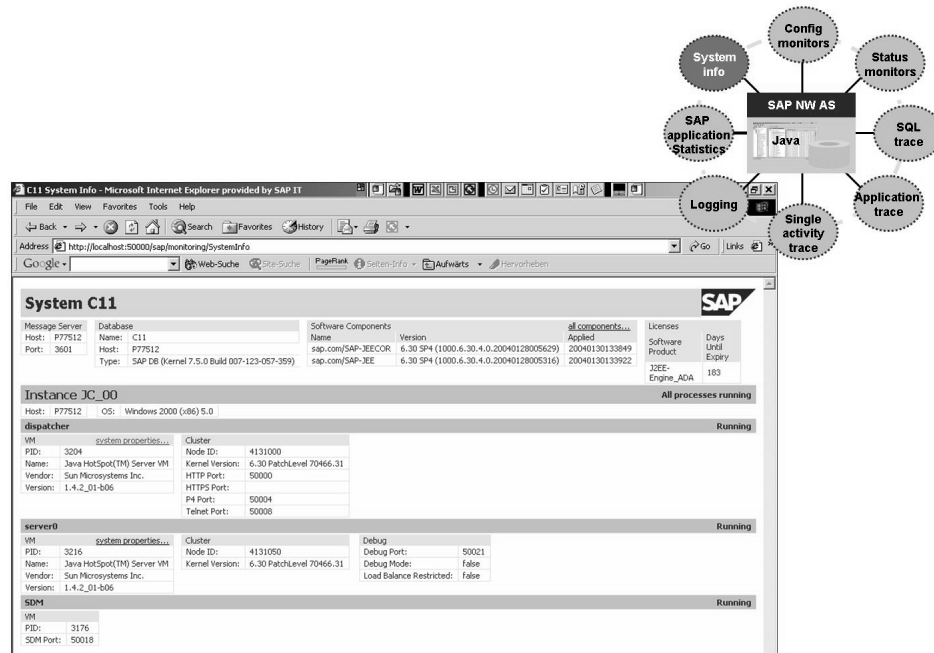


Figure 143: System Info

Integrating the SAP NetWeaver AS Java Monitoring Data into a Central Monitoring System

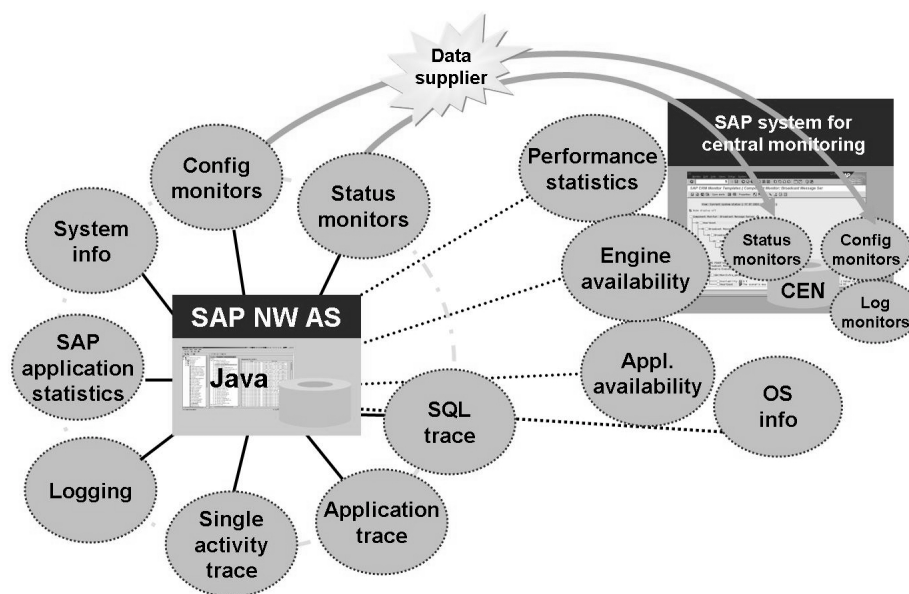


Figure 144: Central Monitoring of SAP NetWeaver AS Java

You can monitor SAP NetWeaver AS Java from a central ABAP monitoring system using the CCMS monitoring architecture. When you do so, the data collected by the JMX monitors is sent to the central monitoring system (CEN) using the CCMS agent SAPCCMSR. In the CCMS Alert Monitor (transaction RZ20), you can display (SAP NetWeaver AS Java) data about availability, status, logs, and JARM (Java Application Response Measurement).

You can view the following SAP NetWeaver AS Java data in the central monitoring system:

- Data from the Monitoring Service
 - Status/configuration monitor and collected performance data
- Data about the availability monitoring of applications and the SAP NetWeaver AS Java
- Statistical data
 - Global Workload Monitor (transaction ST03G)
 - The Global Workload Monitor (transaction ST03G) displays Java statistical records that are used for performance monitoring.

The SAP NetWeaver AS Java can write distributed statistics records (DSRs). DSRs can trace actions that are processed using non-ABAP components such as SAP NetWeaver AS Java, SAP Business Connector (BC), and the Internet Transaction Server (ITS). You obtain an exact picture of the quality of a system as a whole and, in the case of a performance bottleneck, a pointer to its cause. The writing of the statistics records is activated when the SAPCCMSR agent is registered and the standard job SAP_COLLECTOR_FOR_NONE_R3_STAT is started in the SAP ABAP system. You can display the DSRs in aggregated views in the relevant CCMS display transactions (transaction ST03G) in the SAP ABAP system.

– Performance Trace (transaction STATTRACE)

If you find errors in the Global Workload Monitor, you can analyze these using the performance trace (also known as the functional trace). The performance trace displays all the collected statistical data in raw form. performance trace provides duration information for the individual modules of the Engine and therefore provides a finer granularity than the statistics records (DSR).

The performance trace (also called the functional trace) can be written, based on the collected DSRs. You can activate it to be able to analyze the performance of the SAP NetWeaver AS Java from a central CCMS monitoring system (transaction STATTRACE) if you have identified anomalies using DSRs in the ABAP monitoring system.

- Monitoring log files

The functions of the CCMS agents include the monitoring of log files. You can use the SAPCCMSR agent to monitor log files for particular search patterns, the last change time, or for their existence. The results are displayed in the Alert Monitor (transaction RZ20). However, the Log Viewer presentation data cannot be transferred to the CCMS.

- Monitoring the operating system

If the data supplier SAPOSCOL is running on the SAP Web AS Java host, the SAPCCMSR agent also automatically transfers the collected operating system data to the ABAP system.



Lesson Summary

You should now be able to:

- List the SAP NetWeaver AS Java monitoring tools
- List the monitors that display data in a central system

Related Information

- service.sap.com/monitoring
- service.sap.com/javamonitorting

Lesson: Monitoring SAP NetWeaver AS Java

Lesson Overview

SAP NetWeaver AS Java provides an infrastructure that makes monitoring data available. You can display this monitoring data directly in the Visual Administrator or in the SAP NetWeaver Administrator. You can also set threshold values for this data there. Threshold values determine the colors with which data is displayed in the monitor.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the monitoring infrastructure
- Display monitoring data in the “Monitoring” service
- Make threshold value settings in the “Monitoring” service

Business Example

You are using an SAP NetWeaver AS Java. Monitoring is important for safeguarding a stable system environment. It allows for some error situations to be identified in advance. SAP NetWeaver AS Java provides an infrastructure that makes monitoring data available. You can display this monitoring data directly in the Visual Administrator or NWA.

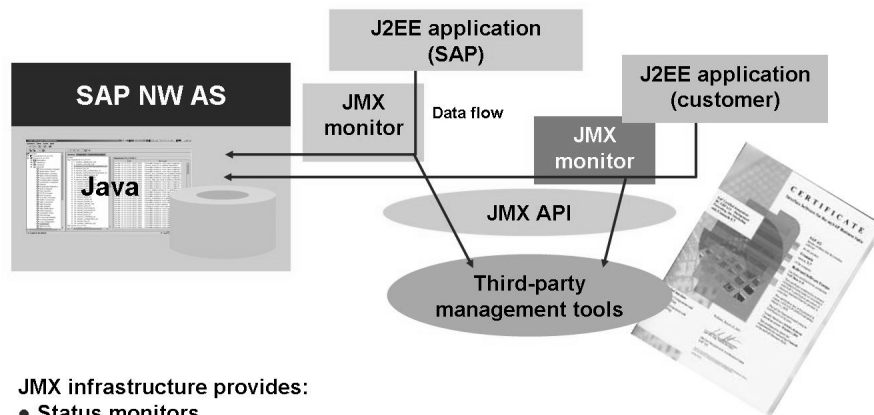
Monitoring Infrastructure

The monitoring in SAP NetWeaver AS Java is based on the standard *Java Management Extension (JMX)*. JMX provides a new flexible administration infrastructure that is used for the monitors in the *Monitoring* service. The JMX infrastructure allows different resources to register as suppliers for monitoring data. Through the JMX API, data is made available for resources of all server components (services, interfaces, libraries, and managers), and applications using MBeans. The data of the JMX monitors is stored in appropriate monitor nodes.



Note: MBean (Manageable Bean):

In the Java programming language, an MBean is a Java object that represents a manageable resource such as an application, service, or component.



JMX infrastructure provides:

- Status monitors
- Configurable monitors

SAP provides customers with:

- SAP templates for JMX monitors to integrate your own J2EE applications

Connection to third-party management tools:

- Display all current values
- Adjust group configurations
- Create/delete groups and install/uninstall monitor nodes

Figure 145: Monitoring Infrastructure (JMX)

Tasks of the JMX interface and the Monitoring service:

- Monitoring the current status
- Creating a history
- Using an alert mechanism to react to critical situations

The JMX infrastructure is provided by the *JMX Adapter* service. Since JMX is a standard, this ensures that external tools can also access the monitoring data. The external tools connect through the JMX API and can display all current values in the JMX monitors. They can also create, delete, and change groups, as well as installing and uninstalling monitor nodes.



Hint: A group gathers monitors together. Each monitors displays individual information for a monitored object.

The data collection is performed at runtime. For this, the data can either be periodically fetched from the JMX monitors (passive), or the resources themselves send the data to the JMX monitors using event mechanisms (active). When the SAP NetWeaver AS Java is started, the JMX monitors are created, and are provided with data at runtime.

The monitoring infrastructure uses XML-based configuration files to create new monitors. SAP delivers the relevant configuration files. If you want to monitor your own J2EE application using JMX, you can use the templates provided by SAP to implement this.

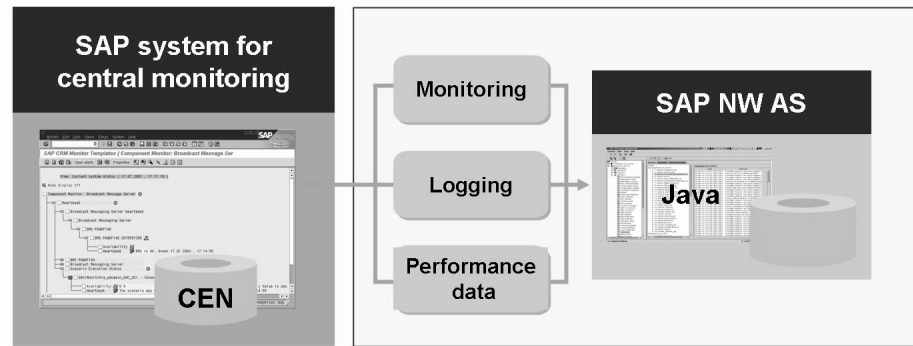


Figure 146: Monitoring - Tools

The collected data is automatically displayed by the Visual Administrator in the *Monitoring* service, or alternatively by NWA in the Monitoring Tree. The data can also be transferred to a central ABAP monitoring system using an SAPCCMSR agent. The next section introduces the *Monitoring* service. Connection to a central ABAP monitoring system is described in one of the following lessons.



Hint: The Monitoring service in the Visual Administrator is a useful monitoring tool in the context of development and test systems. For production systems, we recommend that you use central monitoring with the standard SAP ABAP monitoring tools.

The Monitoring Tree in SAP NetWeaver AS Java

The critical values for the individual managers and services of SAP NetWeaver AS Java, such as memory usage, pool utilization, queue sizes are displayed with alert colors in accordance with the “traffic light system”. In the SAP NetWeaver Administrator (abbreviation: NWA), the data is displayed in the Monitoring Browser. As can be seen in the figure “Monitoring Browser in NWA”, the Monitoring Browser in the reports is located in the path *System Management* → *Monitoring* → *Java System Reports*. You can use *Predefined Local J2EE Views* to switch between the full view and various other views restricted to individual areas of the tree.

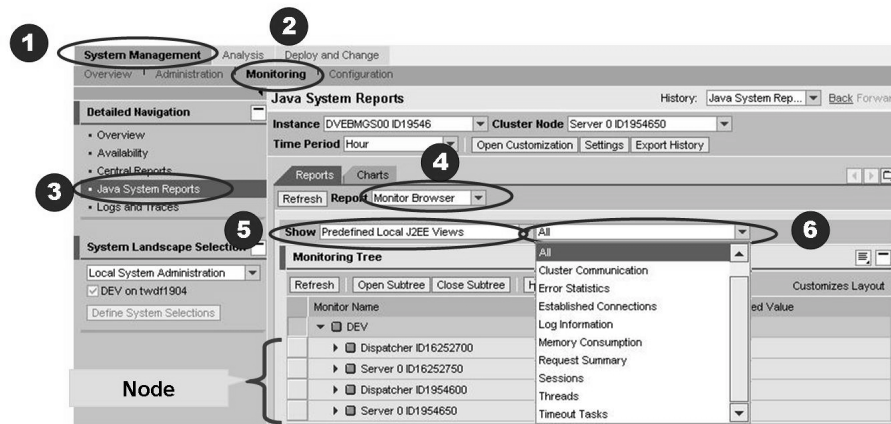


Figure 147: Monitoring Browser in the NWA

In the NWA, the Monitoring Browser displays all the running nodes in the system.

The monitoring function is designed as a tree structure (see figure “Monitoring Tree in the NWA”). Each monitor represents an agent, which communicates with the specific resource and displays the collected data. A monitor usually only shows a small quantity of data, which displays the information about one single aspect of the monitored object. The monitors are combined into monitor groups for clarity. The current values can be read behind each monitor.

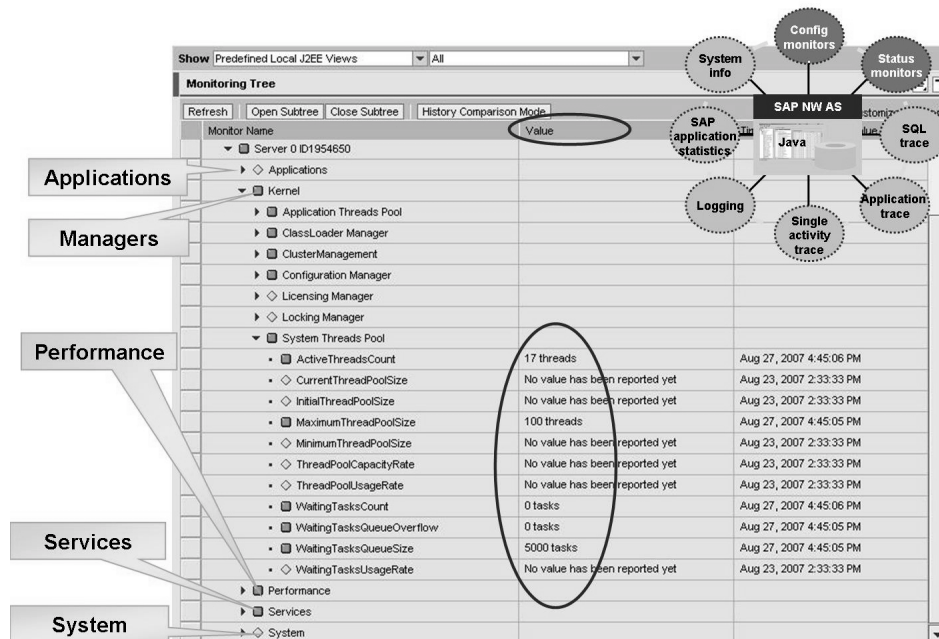


Figure 148: Monitoring Tree in the NWA

The data displayed here can also be transferred to the CCMS of the central system by the SAPCCMSR agent and displayed in the Alert Monitor (transaction RZ20). You can set up additional notifications in the case of alerts and auto-reaction methods there.

In the Visual Administrator, the Monitoring Tree is distributed across the individual nodes, i.e. each monitoring service displays only values for the node under which it is running. The figure “Visual Administrator – Monitoring Service” depicts the entries in the tree structure for a node, in this case a node for a server process.

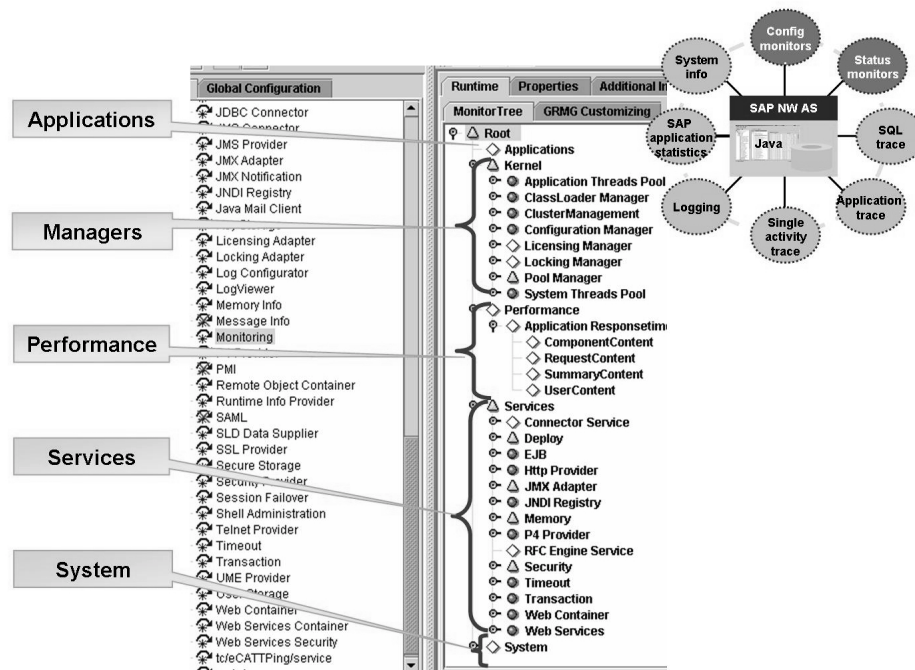


Figure 149: Visual Administrator – Monitoring Service

The tree structure contains the following entries:

- Kernel
Status information for the managers registered for monitoring is displayed under the Kernel entry.
- performance
The Performance area displays available data about performance measurements of the SAP NetWeaver AS Java.
- Services
Status information for the services registered for monitoring is displayed under the Services entry.
- System
The system properties are displayed here.
- Applications
This branch contains information about the status of applications that are running on the SAP NetWeaver AS Java and for which monitoring functions are implemented in the coding. This is a configurable type of monitor, since you can specify which information is displayed in the monitor for your own applications. An application developer usually creates his or her own monitors and objects under the *Applications* branch. The other monitor branches, such as Kernel, System, and so on are reserved for data that is directly and automatically collected by the system.

By default, the monitor *Table Buffer* is always displayed in the *Applications* area along with other items.

In the monitor itself, the statuses are identified with different colors. A color changes when a value exceeds or falls below a threshold value. Errors are highlighted in red and passed on to the highest level of the monitor. You can find the alert that has occurred by expanding the monitor. The following colors can be displayed in the monitor:

- green: According to the settings, the collected values are OK.
- yellow: The collected values have exceeded a predefined threshold value. A warning is displayed.
- red: An error is displayed. Another threshold value has been exceeded.
- white: This has two meanings:
 - The monitor does not contain any performance values; only static information is displayed.
 - The monitor is not functioning correctly.

Activating Monitors, Threshold Maintenance and History Displays in the NWA

When you fully expand a monitor branch in the Monitoring Browser, you see the current values in the *Current Values* tab at the lowest monitor level. Furthermore, as can be seen in the figure “Configuration in NWA”, it is possible to maintain the thresholds and configure data collection in the *Configuration* tab. In the figure, a monitor with a deactivated data collection method is depicted. Consequently, no “History” tab is visible in the figure.

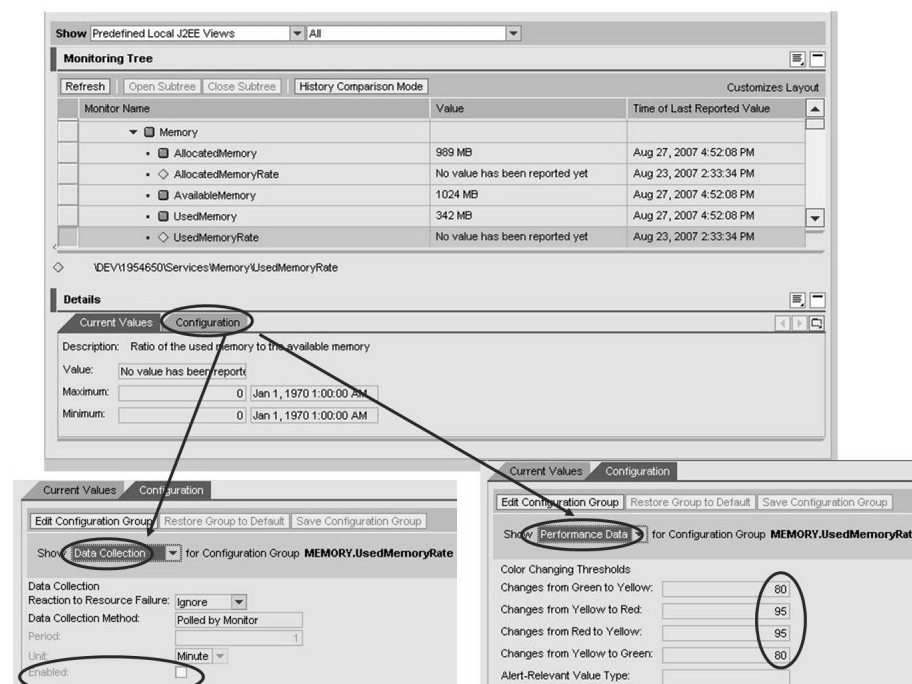


Figure 150: Configuration in the NWA

You can activate data collection methods by checking the box next to the *Enabled* field and saving the configuration. You can set the threshold values in the *Performance Data* view in the *Configuration* tab. A threshold value determines when which alert (color in the monitor) is to be triggered. For a working monitoring that is individually adjusted to your system, you should adjust the threshold values. . The *History* tab (which is not depicted in the figure above) provides a graphic presentation of the most recent collected values subdivided by minutes, quarter hours or hours.

Alongside the history for an individual monitor, it is also possible to compare multiple monitors. See the figure “Comparative History in the NWA”.

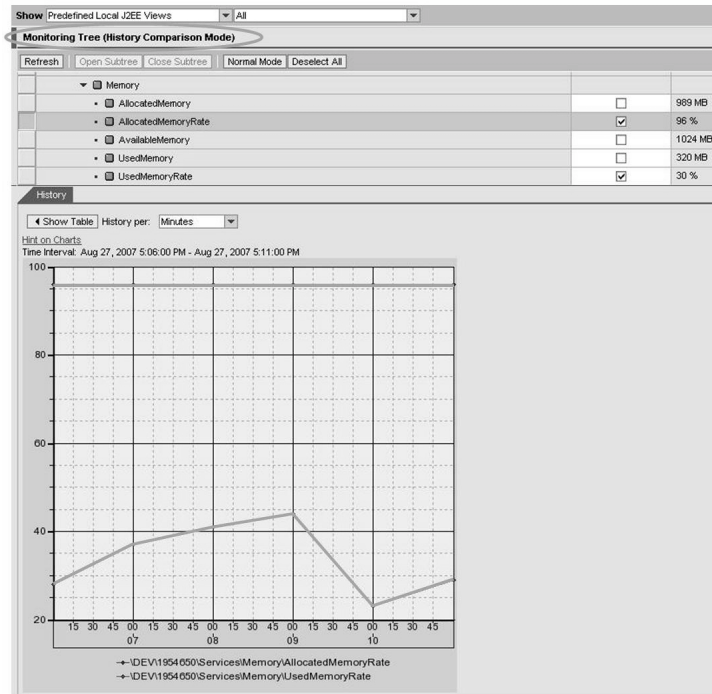


Figure 151: Comparative History in the NWA

In the Monitor Browser, you can use the *History Comparison Mode* button to go to the comparison view. Here, you can select a number of different monitors whose values are compared in either graphical or tabular form. The period for comparison can be set to minutes, quarter hours or hours. In most cases, the comparative history is better used in combination with a restricted view such as *Memory Consumption* than with the full view.

Maintaining Threshold Values and Displaying Histories in the Visual Administrator

When you fully expand a monitor branch in the Monitoring service, a history and the option to configure the threshold values are displayed at the bottom of the monitor.

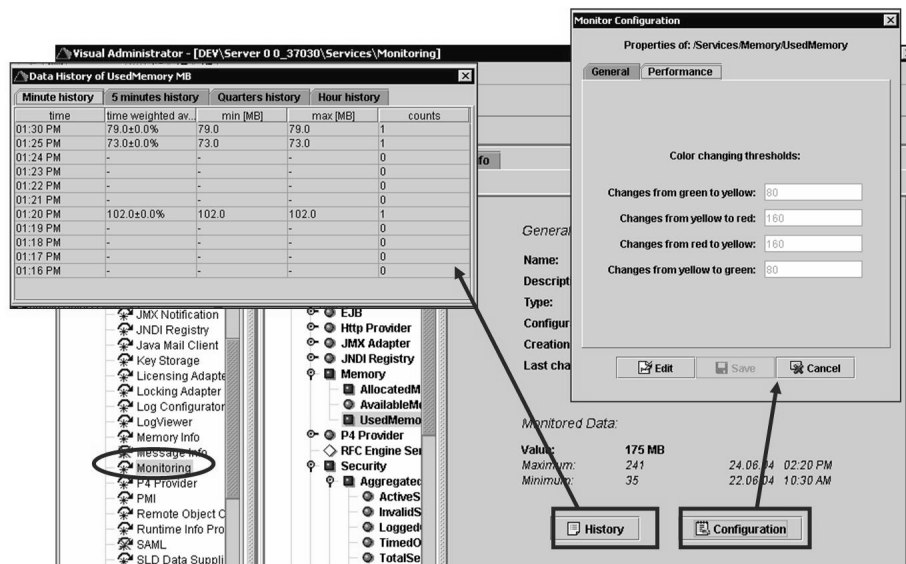


Figure 152: History and Configuration

The **History** contains the latest collected values for each monitor. It also displays, for example, values for each hour. A threshold value determines when which alert (color in the monitor) is to be triggered. For a working monitoring that is individually adjusted to your system, you should adjust the threshold values. You can set threshold values by choosing the *Configuration* icon.

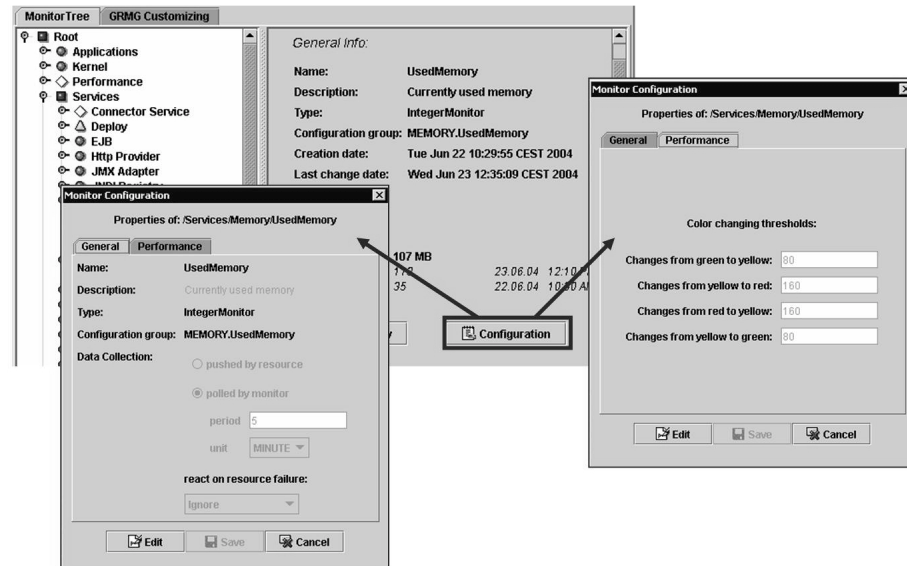


Figure 153: Threshold Values and Data Collection

You can change the frequency of the data collection and the description on the *General* tab page. The *Performance* tab page is visible if you have selected a Performance monitor. You use this tab page to adjust the threshold values. Your changes are stored in the database. The tab page *States* is displayed for static monitors.



Note: To successfully configure monitoring, you require experience in the area of configuration and tuning.

Monitoring Service: Classification

The Monitoring service provides a set of different monitors. You do not need to treat all of these with the same priority.

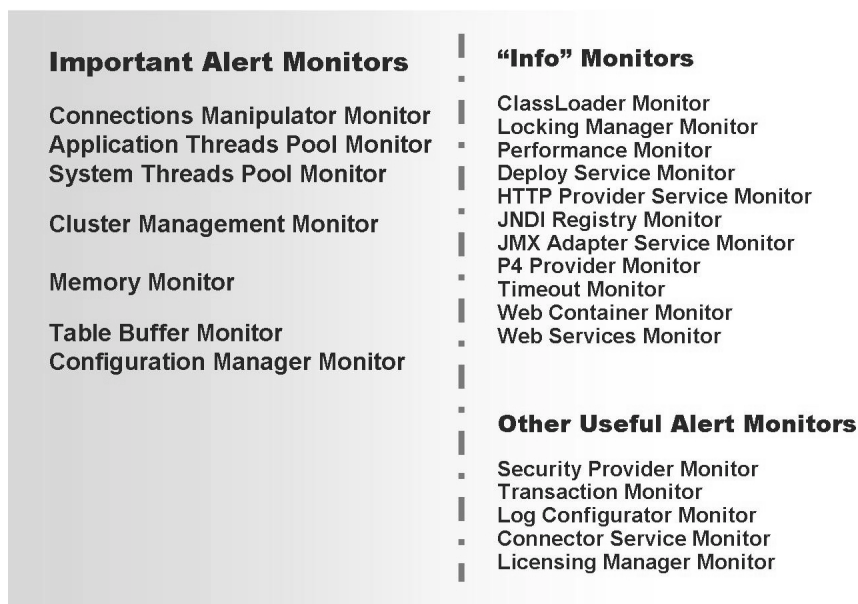


Figure 154: Monitoring Service: Classification

You should pay attention to the monitors that are shown in the figure with the classification *Important Monitors*. You can find information in these about general communication, the processing of requests, and the database connection of SAP NetWeaver AS Java.

The *Other Useful Monitors* area is useful in specific situations. The Log Configurator is often used if you have written your own applications for SAP NetWeaver AS Java and want to monitor log files for it. The Licensing Manager is responsible for the licenses in the system. As soon as you import a permanent license, no alarm will be shown any longer.

The *Info Monitors* are usually of greater importance to a developer than for an administrator.

Exercise 19: Monitoring SAP NetWeaver AS Java

Exercise Objectives

After completing this exercise, you will be able to:

- Monitor the SAP NetWeaver AS Java using the SAP NetWeaver Administrator
- Monitor the SAP NetWeaver AS Java using the Visual Administrator
- Make threshold value settings for individual objects in the monitor

Business Example

For successful monitoring using the Monitoring service in SAP NetWeaver AS Java, you must set the threshold values appropriately.

Task 1: Making Settings with the NWA

Check whether an alert has occurred in the *UsedMemoryRate* in the Memory service of the server processes. If necessary, activate data collection for the *UsedMemoryRate*. Change the threshold value in a server process's Memory service so that a red alert will be displayed in the *UsedMemoryRate* area when 90% of memory is used (yellow: 75%).

1. Log on to the SAP NetWeaver Administrator, open the Monitoring Browser, and check whether an alert has occurred in the Memory service.
2. Check whether data collection is activated for the *UsedMemoryRate* monitor and activate it if necessary.
3. Set the alerting for the *UsedMemoryRate* area so that a red alert is displayed for 90% (yellow: 75).

Task 2: Optional: Making Settings with the Visual Administrator

Check whether an alert has occurred in the Memory service. Change the threshold value in the Memory service so that a red alert will be displayed in the *UsedMemory* area when 250 MB of memory is used (yellow: 200 MB).

1. Log on to the Visual Administrator, open the Monitoring service, and check whether an alert has occurred in the Memory service.

Continued on next page

2. Change the threshold value for the Memory service in the Monitoring service. Set the alerting for the *UsedMemory* area so that a red alert is displayed at 250 MB (yellow: 200 MB).

Solution 19: Monitoring SAP NetWeaver AS Java

Task 1: Making Settings with the NWA

Check whether an alert has occurred in the *UsedMemoryRate* in the Memory service of the server processes. If necessary, activate data collection for the *UsedMemoryRate*. Change the threshold value in a server process's Memory service so that a red alert will be displayed in the *UsedMemoryRate* area when 90% of memory is used (yellow: 75%).

1. Log on to the SAP NetWeaver Administrator, open the Monitoring Browser, and check whether an alert has occurred in the Memory service.
 - a) Log on to your system's SAP NetWeaver Administrator, e.g. <http://<hostname>:<Port>/nwa>. Log on with a user and password (your instructor will provide the user and password information). Go to the Monitoring Browser *System Management* → *Monitoring* → *Java System Reports* → *Report: Monitoring Browser* → *Show: Predefined Local J2EE Views* → *All*.
 - b) Select the appropriate server process and open *Services* → *Memory* there. Here you can see the various monitors. You can use the colors (red, yellow, green, gray) to identify whether an alert has occurred. Navigate to the *UsedMemoryRate* monitor and select the monitor's current values. If you see *No value has been reported yet* then this indicates that data collection is not active. Move on to the next step.
2. Check whether data collection is activated for the *UsedMemoryRate* monitor and activate it if necessary.
 - a) Select *UsedMemoryRate* if you have not already done so.
 - b) Select the *Configuration* and then choose *Data Collection* in the *Show*. Check whether the checkbox next to *Enabled* is selected. If it is not, switch to Editor mode and select the *Edit Configuration Group* button. Now check the box and save your configuration. Click on the *Refresh* button below the “Monitoring Tree” to update the Monitoring Tree.

Continued on next page

3. Set the alerting for the *UsedMemoryRate* area so that a red alert is displayed for 90% (yellow: 75).
 - a) Go back to the *Configuration* tab page. Choose *Performance Data* here.
 - b) If you have not yet done so, switch to Editor mode and select the *Edit Configuration Group* button.

Enter the following values, for example, and save your configuration:

- Green to yellow: 75
- Yellow to red: 90
- Red to yellow: 85
- Yellow to green: 70

Task 2: Optional: Making Settings with the Visual Administrator

Check whether an alert has occurred in the Memory service. Change the threshold value in the Memory service so that a red alert will be displayed in the *UsedMemory* area when 250 MB of memory is used (yellow: 200 MB).

1. Log on to the Visual Administrator, open the Monitoring service, and check whether an alert has occurred in the Memory service.
 - a) Start the Visual Administrator from the directory *G:\usr\sap\<SID>\<Instanz>\j2ee\admin* with the executable file “go.bat”. Log on with a user and password and the appropriate port (your instructor will provide the user, password, and port information).
 - b) Select the appropriate server and open the *Monitoring* service there. The various monitors are displayed on the tab page *runtime* → *Monitor Tree*. You can use the colors (red, yellow, green, gray) to identify whether an alert has occurred. Navigate to the Memory monitor under Services.

Continued on next page

2. Change the threshold value for the Memory service in the Monitoring service. Set the alerting for the *UsedMemory* area so that a red alert is displayed at 250 MB (yellow: 200 MB).
 - a) Open the monitor for Memory as described under 1.b). Select the *UsedMemory* entry.
 - b) Move to the threshold value maintenance by choosing the *Configuration* button and selecting the *Performance* tab page. Switch to change mode and enter the following values, for example:
 - Green to yellow: 200
 - Yellow to red: 250
 - Red to yellow: 230
 - Yellow to green: 180



Lesson Summary

You should now be able to:

- Describe the monitoring infrastructure
- Display monitoring data in the “Monitoring” service
- Make threshold value settings in the “Monitoring” service

Lesson: Appendix: Background Information About the Monitoring Service

Lesson Overview

There are a large number of monitors in the Monitoring service, some of which are especially important. This lesson focuses on the monitors that display data about processing a client request.



Lesson Objectives

After completing this lesson, you will be able to:

- List the most important monitors in the Monitoring service
- Define which managers are involved in processing a request

Business Example

You are using an SAP NetWeaver AS Java. Monitoring is important for safeguarding a stable system environment. SAP NetWeaver AS Java makes monitoring data available in its monitoring infrastructure. You can display this monitoring data directly in the Visual Administrator.

Appendix: Explanations of the Important Monitors

This lesson provides additional background information about the most important monitors in the Monitoring service.

To understand the information displayed by the “important” monitors in the Monitoring service, it is useful to first understand the processing of requests in the dispatcher and in a server.

Background Information: Request Processing - Dispatcher

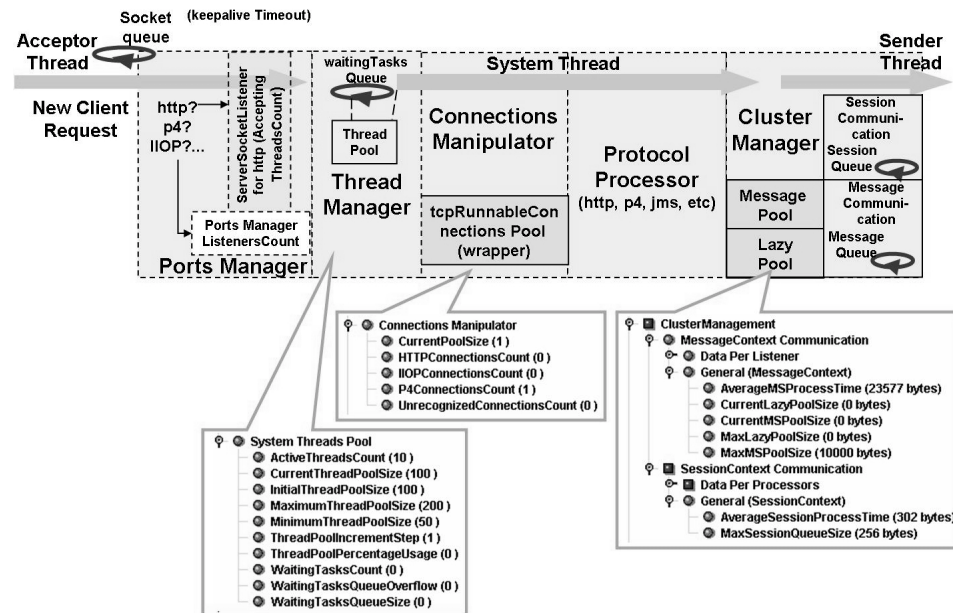


Figure 155: Appendix: Request Processing - Dispatcher

A user sends a new request by HTTP. This request is initially forwarded to the dispatcher. Each dispatcher node has a service with the name *HTTP Provider*. The HTTP provider opens a server socket on the HTTP port to “*ServerSocketListeners*” that are responsible for the requests using HTTP connections. A *ServerSocketListener* itself is responsible for internal scaling in the case of, for example, 1000 simultaneous requests. The task of the *ServerSocketListener* is to create a connection between the client and the dispatcher. By default, SAP NetWeaver AS Java starts 10 *ServerSocketListeners*, which can process approximately 650 new connections per second. If the number of incoming connections is greater than this, the requests are temporarily stored in the HTTP socket queue, and may stay there (if not transmitted) until they receive a keepalive timeout. If the HTTP socket queue is full, the incoming requests are refused and the client must resubmit its request. You can change the following properties in the *HTTP Provider* service: *AcceptingThreadsCount* (for *ServerSocketListener*) and *SocketQueue* (for *HTTP Socket Queue*).

In the next step, the HTTP requests that were forwarded by the *ServerSocketListener* are transferred to a system thread for further processing. Requests that cannot be immediately executed are placed in the *WaitingTaskQueue* of the System Thread Pool of the Thread Manager.

The data is then transferred to the server by the Cluster Manager. In the case of larger data volumes, the data is transferred directly. In the case of smaller data volumes, communication is performed via the message server. The lazy threshold property in the Cluster Manager defines when each type of communication is selected. The Cluster Manager also provides session handling.

Background Information: Request Processing - Server

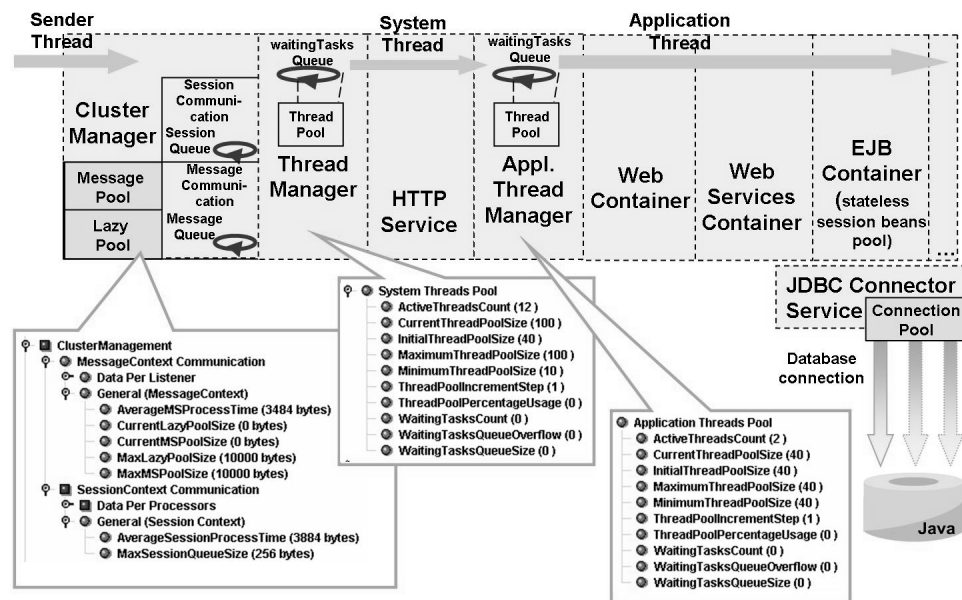


Figure 156: Appendix: Request Processing - Server

The dispatcher communicates with the server through the Cluster Manager. The server starts a system thread and transfers the HTTP request to an application thread. The source code is executed in the application thread. If there are no free application threads, the request is placed in the wait queue of the thread pool. By default, the system can run 40 application threads simultaneously.

Monitor icon: System Thread Pool

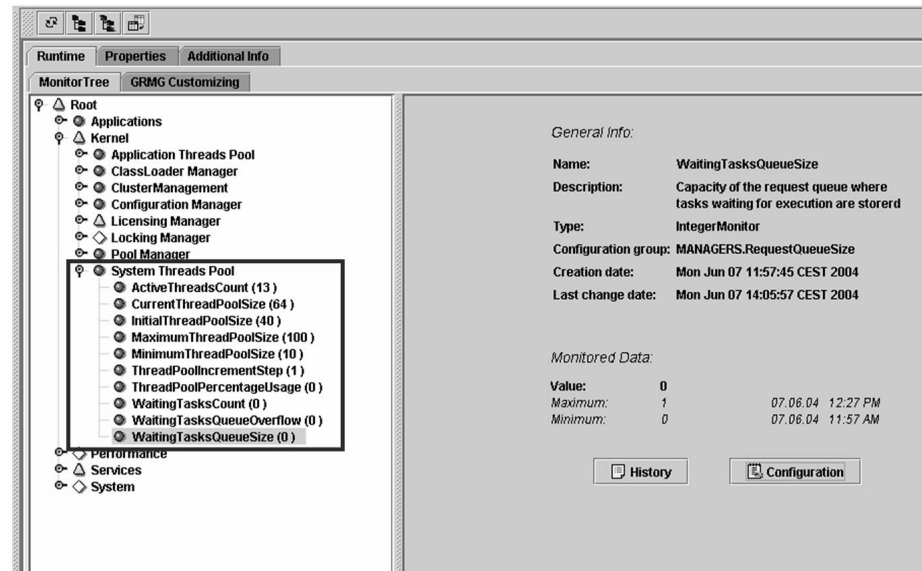


Figure 157: Appendix: Monitor for the System Thread Pool

The *System Thread Pool* exists both for the dispatcher and for the server. The Thread Manager transfers values to the System Thread Pool. The System Thread Pool is responsible for system activities such as backup and background optimization for loaded and held data. You do not usually need to adjust the server threads, since the time of the load is very short. By default, the value 100 is set.

Monitor Area	Description
ActiveThreadsCount	Number of threads from the Thread Pool that are executing a request
CurrentThreadPoolSize	Current number of threads that have been created in the Thread Pool
InitialThreadPoolSize	Initial size of the thread pool
MaximumThreadPoolSize	Maximum size of the thread pool
MinimumThreadPoolSize	Minimum size of the thread pool
ThreadPoolIncrementStep	If the Thread Pool is almost exhausted, it is extended by a certain number of threads. This number is displayed here.
ThreadPoolPercentageUsage	Current usage of the Thread Pool as a percentage

WaitingTasksCount	Number of requests that are waiting for a free thread from the Thread Pool
WaitingTasksQueueOverflow	Number of threads that are waiting to place a request in the request queue (if it is full)
WaitingTasksQueueSize	Size of the request queue

Monitor icon: Application Thread Pool

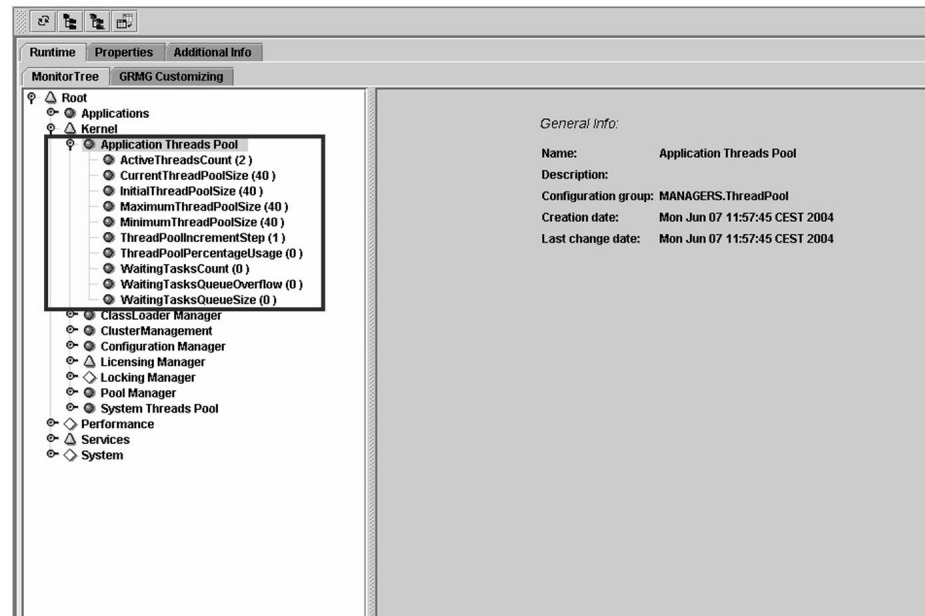


Figure 158: Appendix: Monitor for Application Thread Pool

The Application Thread Manager supports the threads in which source code must be executed. When an HTTP request reaches SAP NetWeaver AS Java this is passed on to an application thread. The Application Thread Pool shows the requests that arrive in the system.

Monitor Area	Description
ActiveThreadsCount	Number of threads from the Thread Pool that are executing a request
CurrentThreadPoolSize	Current number of threads that have been created in the Thread Pool
InitialThreadPoolSize	Initial size of the thread pool

MaximumThreadPoolSize	Maximum size of the thread pool
ThreadPoolIncrementStep	If the Thread Pool is almost exhausted, it is extended by a certain number of threads. This number is displayed here.
ThreadPoolPercentageUsage	Current usage of the Thread Pool as a percentage
WaitingTasksCount	Number of requests that are waiting for a free thread from the Thread Pool
WaitingTasksQueueOverflow	Number of threads that are waiting to place a request in the request queue (if it is full)
WaitingTasksQueueSize	Size of the request queue

Monitor icon: Cluster Management

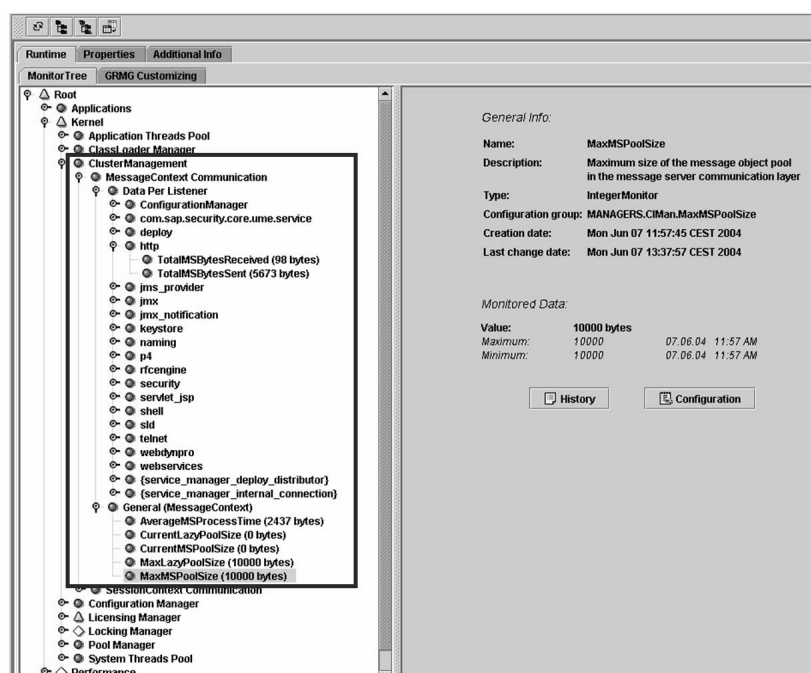


Figure 159: Appendix: Monitor for Cluster Management

The Cluster Manager is responsible for the communication within the SAP NetWeaver AS Java cluster. It is used to exchange messages between the cluster elements. Every cluster node has a connection to the message server with which short messages can be exchanged. Special services require server to server communication, which is always

performed using the message server for small quantities of data. If the quantity of data is larger, a direct connection is provided. The lazy threshold parameter determines when each type of communication is used.

Monitor Area:	Description
MessageContextCommunication	
TotalMSBytesReceived	Number of bytes that was received by a service using the server communication layer
TotalMSBytesSent	Number of bytes that was sent by a service using the server communication layer
AverageMSProcessTime	Average time in milliseconds that a message in the message server area took

Monitor Area:	Description
LazyContextCommunication	
CurrentLazyPoolSize	Current size of the message object pool in the lazy communication area
CurrentMSPoolSize	Current size of the message object pool in the message server area
MaxLazyPoolSize	Maximum size of the message object pool in the lazy communication area
MaxMSPoolSize	Maximum size of the message object pool in the message server area

Monitor Area:	Description
SessionContextCommunication	
CurrentSessionQueueSize	Current size of the message queue in the session communication area
TotalSessionBytesReceived	Number of bytes that was received by a service using the session communication layer

TotalSessionBytesSent	Number of bytes that was sent by a service using the session communication layer
AverageSessionProcessTime	Average time in milliseconds that a message in the communication area took
MaxSessionQueueSize	Maximum size of the message queue in the session communication area

Monitor icon: Memory Service

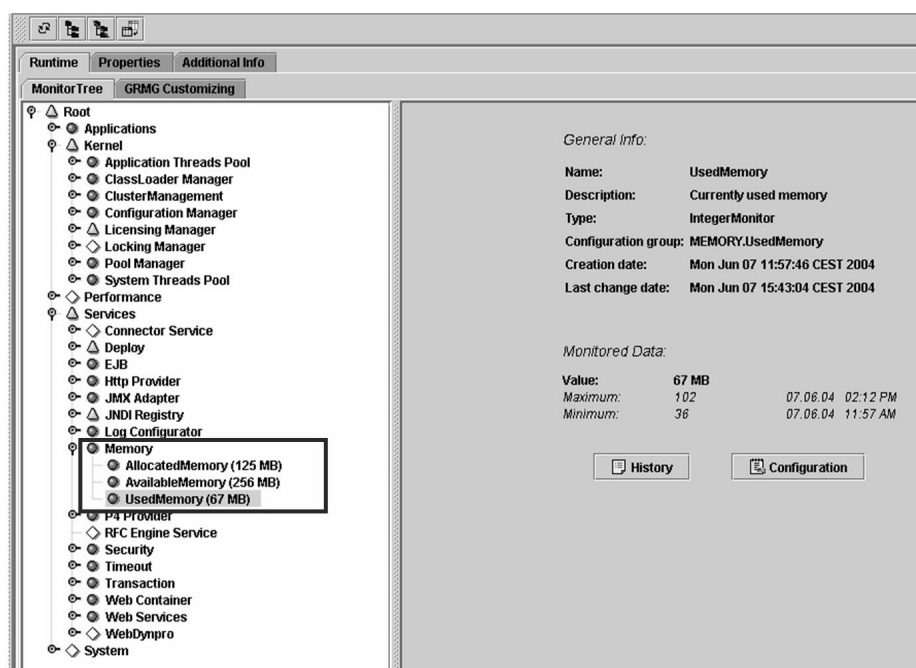


Figure 160: Appendix: Monitor for Memory Service

For a high-performance SAP NetWeaver AS Java that is under load, it is important to keep the available memory and the number of requests processed simultaneously on a server balanced. For this reason, there is a Memory service and two thread managers (the System Thread Manager and the Application Thread Manager). The Memory service is used to observe the memory used internally by the JVM of the owner cluster element. To save memory, the service sends events (session expiration, and so on) to managers and services.

Monitor Area	Description
AvailableMemory	Total memory that can be used by the JVM (Java parameter -Xmx)
AllocatedMemory	Part of the available memory that is actually to be used next, and also the part that is already in use
UsedMemory	Overview of the used memory

Monitor icon: Table Buffers

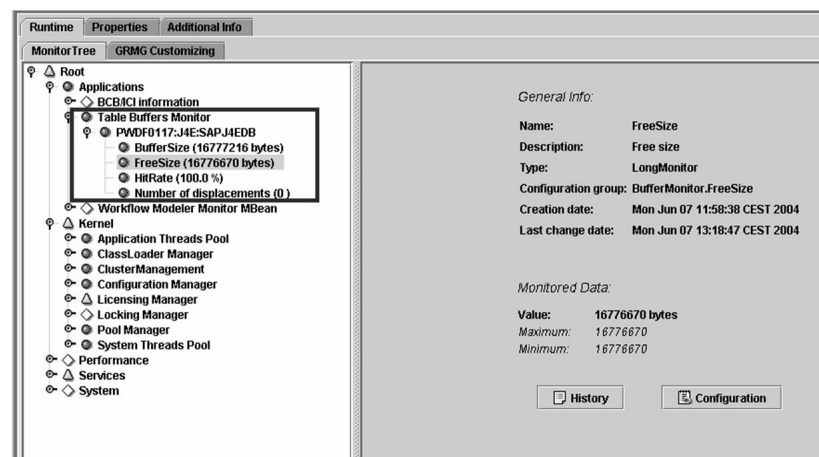


Figure 161: Appendix: Monitor for Table Buffers

Table buffering is one of the most important performance-related features of Open SQL. A separate table buffer is maintained for each server of the cluster. The table buffer stands between the application and the database. To reduce database load and network communication, parts of a database table are loaded into the cache. Individual tables can be marked for buffering using the Java Dictionary.

Monitor Area	Description
BufferSize	Maximum size of the table buffer
FreeSize	Free memory space in bytes
HitRate	Hit rate (requests - buffer)
Number of displacements	Total number of displacements from the buffer since the instance was started

Monitor icon: Configuration Manager

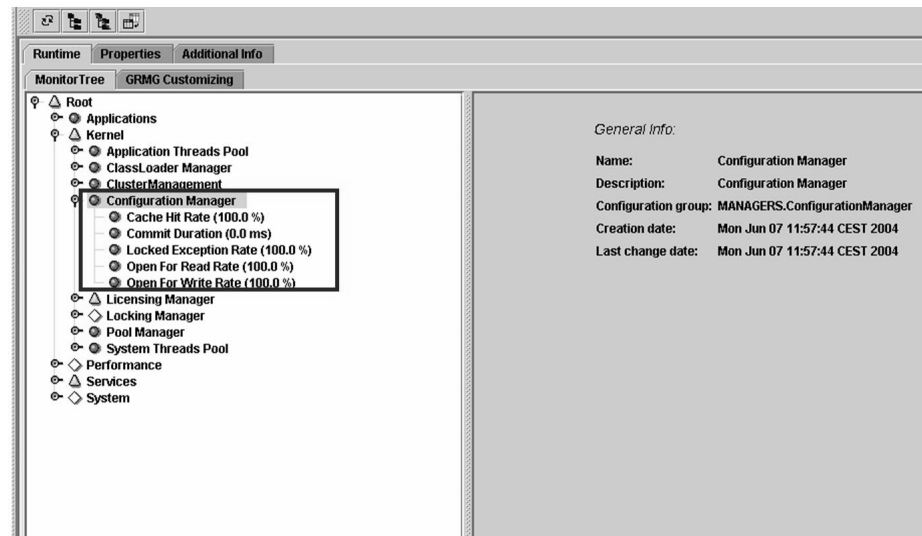


Figure 162: Appendix: Monitor for Configuration Manager

The Configuration Manager allows modules of SAP NetWeaver AS Java to store or access data in a relational database system (RDBMS).



Lesson Summary

You should now be able to:

- List the most important monitors in the Monitoring service
- Define which managers are involved in processing a request

Related Information

- service.sap.com/monitoring
- service.sap.com/javamonitoring

Lesson: Connecting to a Central Monitoring System

Lesson Overview

You can monitor SAP NetWeaver AS Java directly with the Monitoring service of the Visual Administrator or using a central monitoring system. To be able to display the data in the central monitoring system, you need to install the SAPCCMSR agent. The new installation routines and the configuration steps are presented here.



Lesson Objectives

After completing this lesson, you will be able to:

- Monitor Java instances in the central monitoring system using an agent
- Install the SAPCCMSR agent for Java instances
- Explain which configuration steps are required to be able to maintain the threshold values for Java instances from the central monitoring system

Business Example

You use a number of SAP systems in your company. You monitor these SAP systems using a central monitoring system. You have now also installed an SAP system with which you are going to use Java functions. You are therefore using an SAP NetWeaver AS Java, which you want to monitor in the central monitoring system, like your other SAP systems. Using an agent, you can display the most important system data in a central SAP ABAP monitoring system.

Transferring Monitoring Data to a Central Monitoring System

On the SAP NetWeaver AS Java, there is a monitoring infrastructure that collects various data and makes this available in the Visual Administrator. You can display this data in a central SAP monitoring system using the SAPCCMSR agent. The agent is part of the monitoring infrastructure of an SAP ABAP system. After a few simple configuration steps, the SAPCCMSR agent is almost automatically installed using new routines.

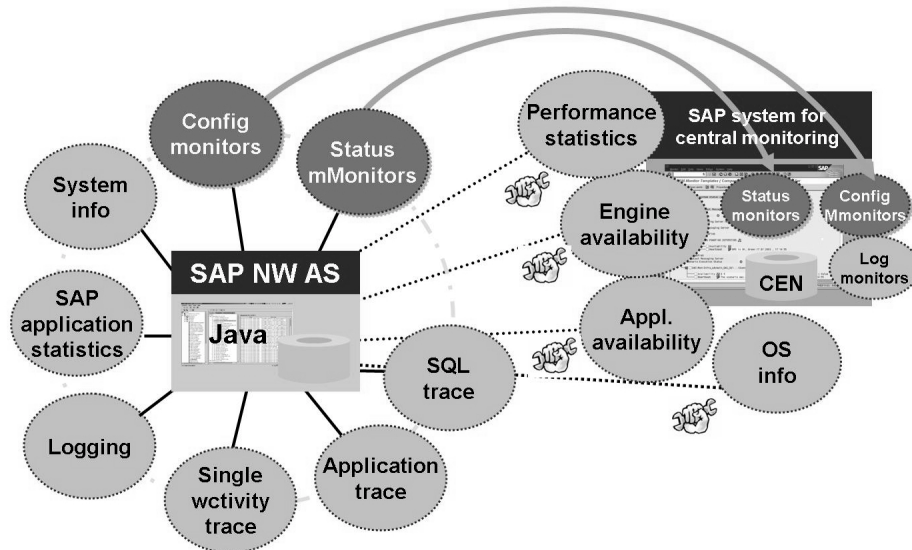


Figure 163: Connecting to a Central SAP ABAP Monitoring System

If the SAP NetWeaver AS Java starts, JMX monitors are created. They deliver data for runtime monitoring. During its installation, the SAPCCMSR agent creates a separate shared memory segment, to which the monitoring data of the SAP NetWeaver AS Java is written. The CCMS Connector routes data from the JMX monitors to the shared memory segment of the agent. An agent always has a connection to the shared memory segment assigned to it and a so-called RFC connection (RFC: remote function call) to the central monitoring system. The agent actively sends the data to the central monitoring system every 60 seconds. Alerts that occur are stored in the database of the central monitoring system. You can display the data either directly in the Alert Monitor (transaction RZ20) or in the SAP Solution Manager.



Hint: In addition to system monitoring, the SAP Solution Manager provides other functions such as the Support Portal, running/ordering SAP Services, Customizing Scout, and others.

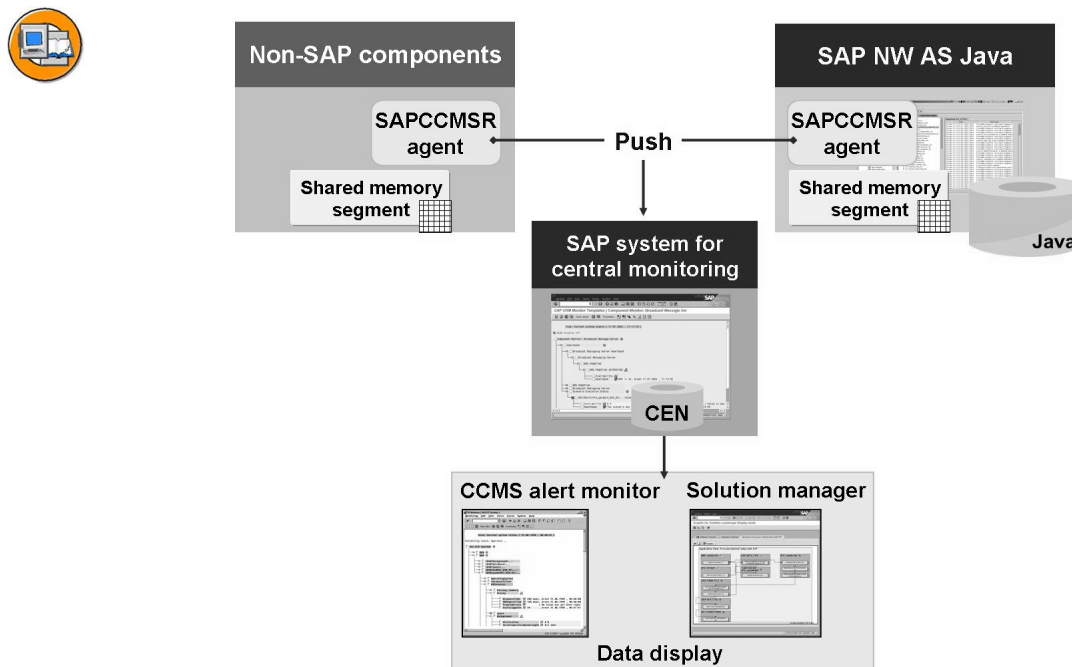


Figure 164: Data Transfer Using the SAPCCMSR Agent

When the SAP NetWeaver AS Java is started, the monitors are created, and are provided with data at runtime. The monitoring data of these monitors is shown in the Alert Monitor (transaction RZ20) of the central ABAP monitoring system. To do this, the monitoring data is forwarded from the SAP NetWeaver AS Java to the shared memory segment of the SAPCCMSR agent. The agent is also required if a Java instance is installed on the same host as the monitoring SAP system.

You can make settings in the central monitoring system that define when which value triggers a red or yellow alert. This is known as maintaining threshold values. For more information, see one of the following sections.

Installation and Configuration of the SAPCCMSR Agent

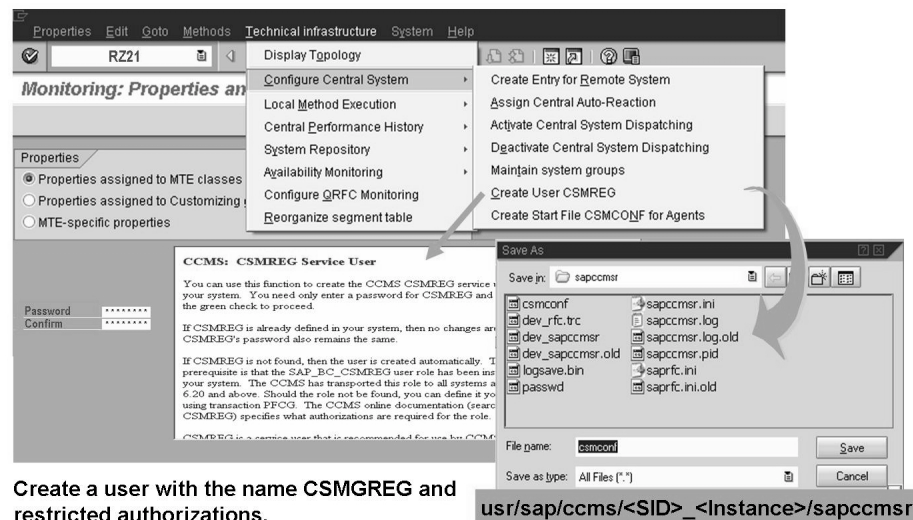
The following steps are required to install the SAPCCMSR agent:

1. Create the CSMREG user in the central monitoring system (transaction RZ21)
2. Create the CSMCONF file in the central monitoring system (transaction RZ21)
3. Register the agent in the Visual Administrator (*Dispatcher* → *Services* → *Monitoring*)

For a dialog-free installation, the agent requires a file called CSMCONF. It is stored in the SAPCCMSR agent working directory `/usr/sap/ccms/<SID_Instance>/sapccmsr`. This file contains users and important system information for the central monitoring system, which are required for the installation. It contains entries about the CSMREG user and an administration user. The CSMREG user is used for communication between the SAPCCMSR agent and the central monitoring system. This user is a communication user with very specific authorizations. The administration user is only used to create the RFC connection from the agent to the central monitoring system; that is, only during the installation itself. The passwords of the users are not stored in the file.



Note: If you have already created the CSMREG user in your system then its previous settings will not be changed.



Create a user with the name CSMREG and restricted authorizations.

`usr/sap/ccms/<SID>_<Instance>/sapccmsr`

CSMCONF:

File containing all information about the central system that the agent requires for registration.

Figure 165: Creating the CMSREG User and the CSMCONF File

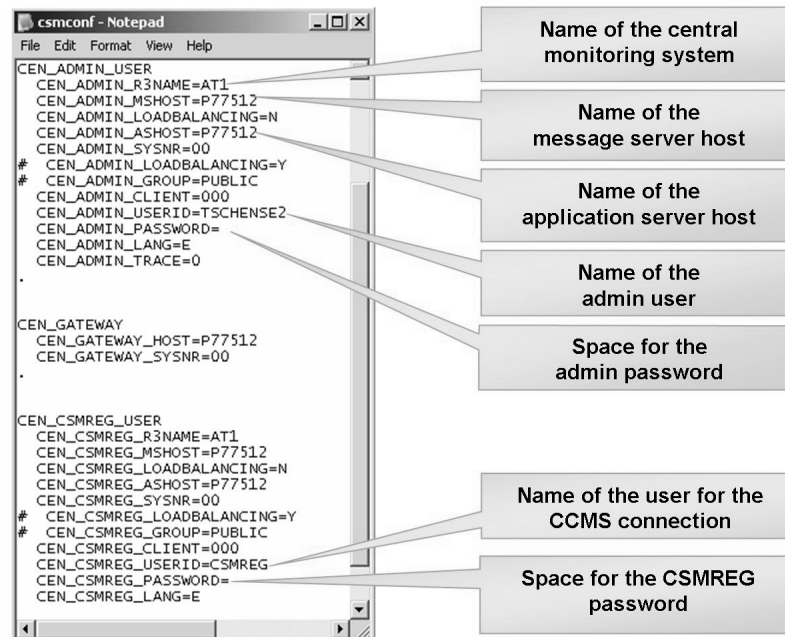


Figure 166: Contents of the CSMCONF

Check whether the executable file for the SAPCCMSR agent is located in the appropriate directory (Microsoft Windows: `[drive]:\usr\sap\<SID>\SYS\exe\run`, UNIX: `/usr/sap/<SID>/SYS/exe/run`, UNIX `/usr/sap/<SID>/SYS/exe/run`).

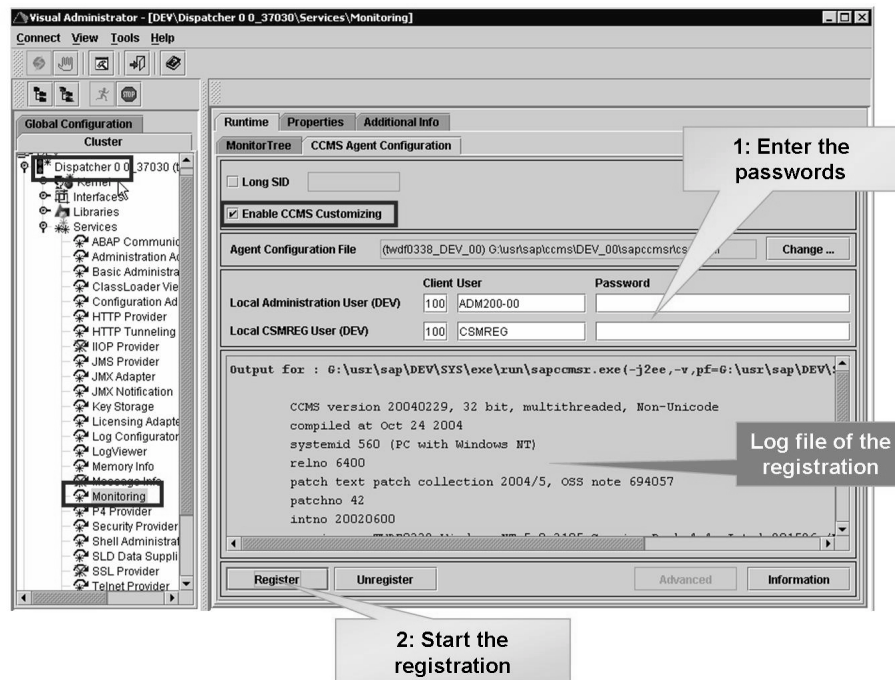


Figure 167: Agent Registration in the Visual Administrator

The SAPCCMSR agent is registration with the Visual Administrator. Navigate in the Visual Administrator to *Dispatcher* → *Services* → *Monitoring* → *Runtime* → *CCMS Agent Configuration*. Leave the *Enable CCMS Customizing* checkbox checked. Verify that the *Agent Configuration File* input field points to where the *csmlconf* file is stored. In the input field *Local Administration User*, enter the password for the user with administration authorization in the central monitoring system with which you created the CSMCONF start file. In the input field *Local CSMREG User*, enter the password for the CSMREG user in the central monitoring system. Choose the *Register* button. Check the registration log. Errors are highlighted in red.



Hint: The agent is automatically started by the startsap script under UNIX when you start the SAP NetWeaver AS Java. In the case of Windows-based systems, the agent is created as a service during installation. After registration via the Visual Administrator, these services must be started.

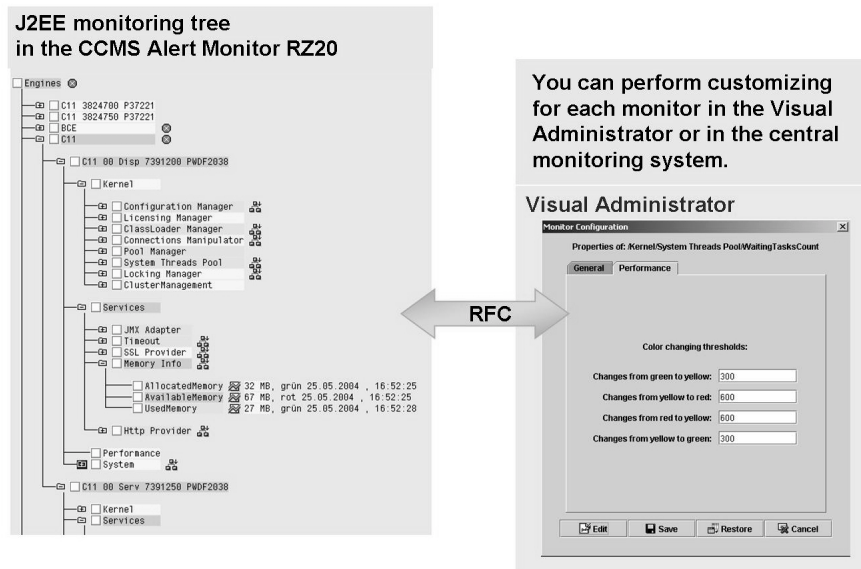
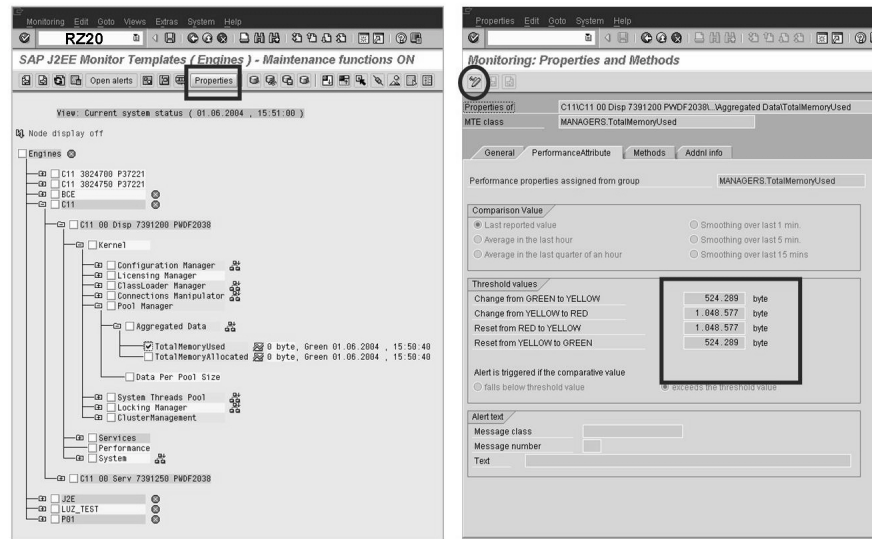


Figure 168: Connection from SAP NetWeaver AS Java to CEN (Threshold Value Maintenance)

The agent allows you to transfer the alerts that have occurred to the central monitoring system. The system should only display an alert if a value exceeds or falls below a specific threshold value, which is entered individually for a system. A threshold value defines the value/status at which an alert with a certain classification (red, yellow, green) is displayed. You can perform this configuration of the threshold values not only in the Visual Administrator, but also in the central monitoring system. When the *enable CCMS Customizing* checkbox is checked, you can change the threshold values of any Java performance node of the monitored system from the CEN.



All monitoring configuration data is persistent and is stored in the Java database.

Figure 169: Threshold Value Maintenance in the Central Monitoring System

You can change the threshold values in the Alert Monitor. Call transaction RZ20, and expand the *SAP J2EE Monitor Templates* monitor set. Start the *Engines* monitor. Expand the tree structure completely, and select, for example, a server node in the central instance in the tree. Now choose the *Properties* button and switch to change mode. You can now maintain its threshold values.

Displaying the Monitoring Data in the Central Monitoring System

You can display the J2EE monitoring data in the central monitoring system using the Alert Monitor. To do this, you must open the Alert Monitor (transaction RZ20) and select the monitor set *SAP J2EE Monitor Templates*. The status data is stored in the following monitors:

- The *Engines* monitor displays status data for the kernel, services, performance, and the system.
- The *Applications* monitor displays application data.

In the SAP NetWeaver AS Java status monitors, you can see at a glance where warnings (yellow) and errors (red) have occurred. If you open the tree at the corresponding places, you learn more about the cause.

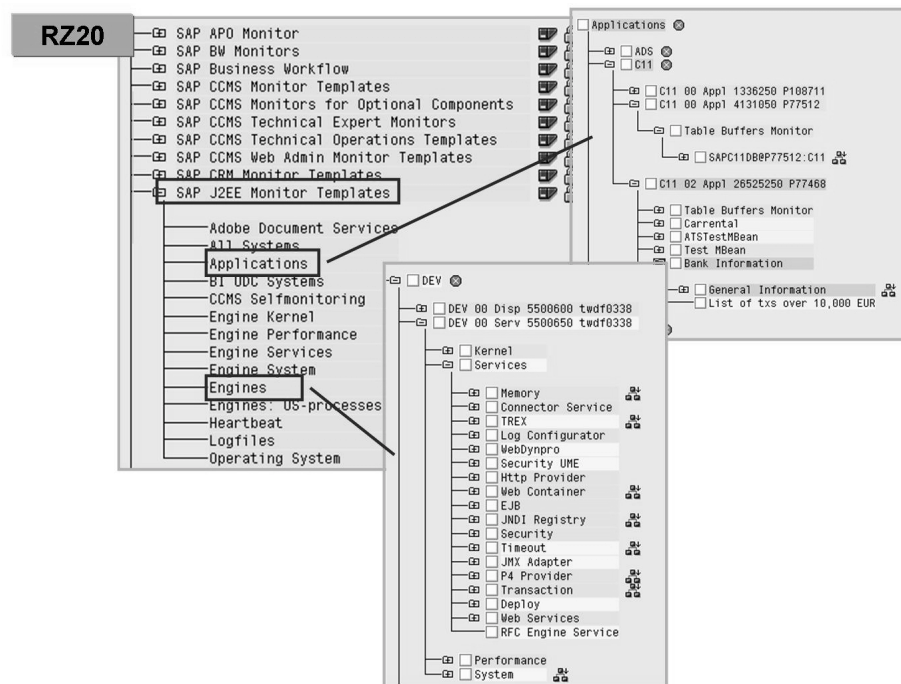


Figure 170: Display in Transaction RZ20

If the Deploy service is colored yellow, the deployment of at least one application has taken so long that a yellow alert was triggered. If you open the tree at this point, you can see which application this concerns. You should check whether the deployment was processed correctly, and that the application is working. You can use the Visual Administrator to check whether the deployment was processed correctly. The application has a checkmark in the Deploy Service under Application, it has been started. The Java Application Response Measurement (JARM) data is displayed under Performance, and operating system data is displayed under System.

The Applications monitor displays monitoring data for J2EE applications that have implemented a monitoring function.

Appendix: Operating System Monitoring in the Central Monitoring System

The operating system information is collected by SAPOSCOL. SAPOSCOL is usually automatically started. The SAPCCMSR agent is required to transfer the monitoring data from the SAP NetWeaver AS Java to the central monitoring system.

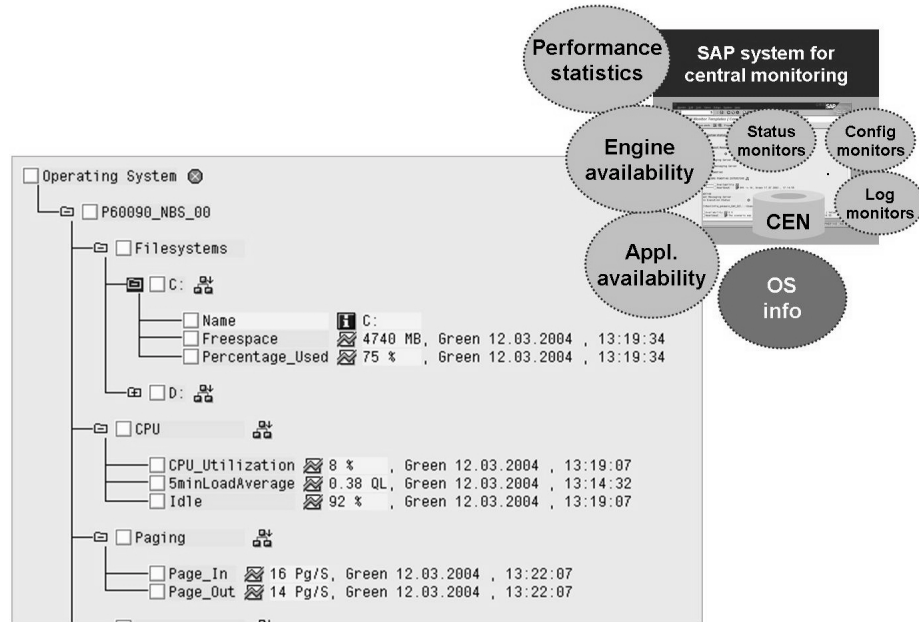


Figure 171: Operating System Information in Transaction RZ20

Exercise 20: Connecting to a Central Monitoring System

Exercise Objectives

After completing this exercise, you will be able to:

- Install an agent that allows you to include an SAP NW AS Java in a central monitoring system

Business Example

You can monitor SAP NW AS Java directly with the Monitoring service of the Visual Administrator or using a central monitoring system. To be able to display the data in the central monitoring system, you need to install the SAPCCMSR agent. The new installation routines and the configuration steps are presented here.

Task: Agent Installation

Install an agent to monitor the Java instances of your SAP NW AS. Check whether the data is displayed in the Alert Monitor (transaction RZ20).

1. Create the CSMREG user and the CSMCONF file. Store the CSMCONF file under the path *G:\usr\sap\CCMS\<system_instance>\sapccmsr*.



Caution: Log on to the SAP system:

Start the SAP GUI for Windows on the operating system of your training host.

2. Start the agent registration in the Visual Administrator (*Dispatcher* → *Services* → *Monitoring* → *Runtime* → *CCMS Agent Configuration*).
3. Start the *sapccmsr* services which belong to your system.
4. Check whether the monitoring data is displayed in the Alert Monitor (transaction RZ20).

Solution 20: Connecting to a Central Monitoring System

Task: Agent Installation

Install an agent to monitor the Java instances of your SAP NW AS. Check whether the data is displayed in the Alert Monitor (transaction RZ20).

1. Create the CSMREG user and the CSMCONF file. Store the CSMCONF file under the path *G:\usr\sap\CCMS\<system_instance>\sapccmsr*.



Caution: Log on to the SAP system:

Start the SAP GUI for Windows on the operating system of your training host.

- a) Log on to your SAP system and start transaction RZ21.
 - b) Follow the instructions from the figure “Creating the CSMREG User and the CSMCONF File”. Store the CSMCONF file under the path *G:\usr\sap\CCMS\<system_instance>\sapccmsr*.
2. Start the agent registration in the Visual Administrator (*Dispatcher → Services → Monitoring → Runtime → CCMS Agent Configuration*).
 - a) Log on to the operating system of the host with the Terminal Services Client
 - b) Start Visual Administrator. Navigate to *Dispatcher → Services → Monitoring → Runtime → CCMS Agent Configuration*.
 - c) Follow the instructions from the figure *Agent Registration in the Visual Administrator*. Enter the passwords and choose the *Register* button.
 3. Start the *sapccmsr* services which belong to your system.
 - a) Open the Services console on the operating system on which your SAP system is running, for example with *Start → run* and execute *services.msc*.
 - b) Search for *sapccmsr.<xx>* where *<xx>* stands for the instance number of your system. Start this, for example, with *right mouse button → Start*.

Continued on next page

4. Check whether the monitoring data is displayed in the Alert Monitor (transaction RZ20).
 - a) Start transaction RZ20 in your SAP system.
 - b) Open the “SAP J2EE Monitor Templates” monitor set and choose the “Engines” monitor (or “Applications”). Start the monitor by double-clicking it. You should now see data, as in the figure “Display in Transaction RZ20”.



Note: It can take a few minutes before the data becomes visible.



Lesson Summary

You should now be able to:

- Monitor Java instances in the central monitoring system using an agent
- Install the SAPCCMSR agent for Java instances
- Explain which configuration steps are required to be able to maintain the threshold values for Java instances from the central monitoring system

Related Information

- service.sap.com/javamonitorting
- service.sap.com/monitoring

Lesson: Log Viewer and Log Configuration

Lesson Overview

Logging and tracing are important functions in the context of error detection. You can configure the level of detail in which information is written to log files. You can access all log files with the Log Viewer.



Lesson Objectives

After completing this lesson, you will be able to:

- Operate the integrated and the central Log Viewer
- Explain the difference between logging and tracing
- Discuss the most important functions of the Log Configurator service
- Use the Log Configurator service to adjust the severity of log files

Business Example

You are working with SAP NetWeaver AS Java and want to know more about the options for configuring and evaluating log files. Since a great deal of log information is created in the SAP NetWeaver AS Java environment, it is important to be familiar with a tool that automatically displays the log files for stable operation.

Log and Trace Files

All Java nodes write log and trace information to files in the file system. These files are formatted in a special way. This formatting makes it possible to use filters to hide or display specific entries when viewing the files in a Log Viewer. The files which possess this formatting are known as “ListLog”s. The entries in the ListLogs also contain a **Severity** field which indicates the weighting of the entry. Some of the ListLogs are listed in the figure “ListLogs in the File System”. For each Java node, i.e. each Java dispatcher and Java server process, there is a separate directory named “log” in the file system under which the files for the node are stored. A basic distinction is made between log and trace files. Log files are sometimes also referred to as logging files. The **trace files** comprise only files with the name **default.<x>.trc** where the <x> stands for a sequential number. The trace files which are discussed here should not be confused with other “trace” files such as the developer traces. The log files include the other files displayed in the figure.

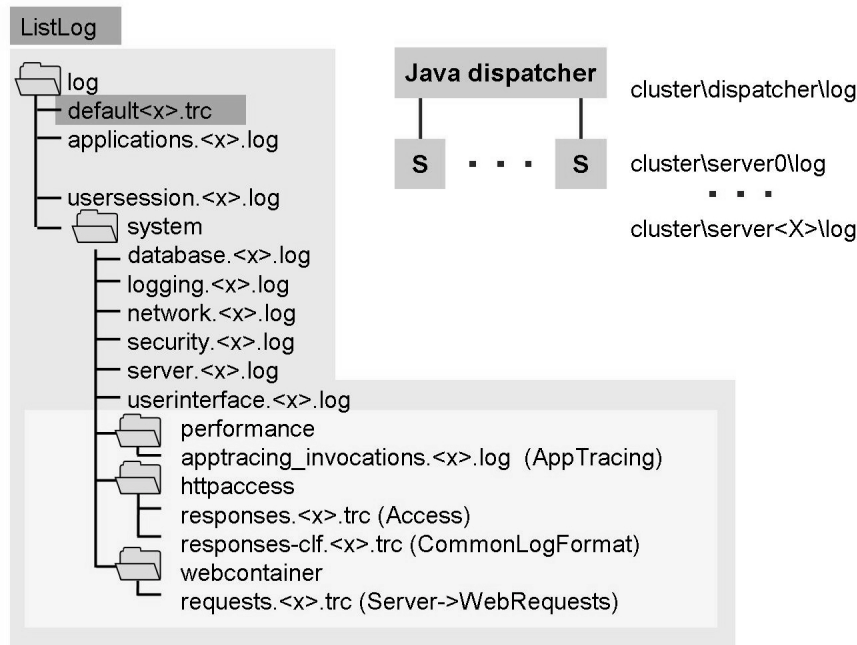


Figure 172: ListLogs in the File System

Java server processes have more log files than Java dispatchers. These are located in a subdirectory of the “system” directory as indicated in color in the figure. The specifications in brackets relate to the log configuration which we will become familiar with later.

Log files are displayed in the Log Viewer. A distinction is made between different types of log file.

The following distinction is made between logging and tracing.

Logging means:



- Recording normal and exceptional events
- Runtime information of a system or an application is written to log files
- Active during normal operation

Tracing means:



- Recording the process flow of an application
- Use during development and for error detection in the production environment
- All traces are stored in the default.<x>.trc files

The Log Viewer

To ensure stable operation, the log and trace files should be regularly checked for error messages.

SAP provides a mechanism for the automatic analysis of log and trace files. You can evaluate and monitor the log files in two ways:

- Central monitoring with SAP NetWeaver AS ABAP

If you are using an SAP NetWeaver AS ABAP that is acting as a central monitoring system, you can also use the standard monitoring methods of the ABAP environment. You can use the CCMS agent SAPCCMSR to search the log files every minute for predefined search patterns. If the agent finds a pattern, it reports an alert in the central monitoring system. The administrator can be notified from there on the basis of the alert.

- Monitoring with the infrastructure of SAP NetWeaver AS Java (Log Viewer, and so on)



Note: The focus of this lesson is on monitoring with SAP NetWeaver AS Java and its infrastructure.

SAP NetWeaver AS Java itself also provides an infrastructure for centrally viewing logs and traces. You can use the central Log Viewer to monitor log files across systems. This is made possible as there is an infrastructure for central logging and tracing. In this case, all Java components use the same logging/tracing infrastructure. The Log Viewer is part of the logging/tracing infrastructure.



Note: The logging/tracing infrastructure is described in more detail in the following sections.

The **Log Viewer** is always used to display log and trace files, irrespective of whether they are created by the kernel, services, libraries, or applications. The log files for all server nodes can be combined. The Log Viewer can search log files for entries that have a specific weighting (severity). You can use the Log Viewer in the following variants:



- **As Log Viewer in the SAP NetWeaver Administrator**
 - Log and trace files for the runtime environment and the running applications are automatically registered
 - Predefined views are supplied
 - You can also save user-defined views
- **As an integrated Log Viewer (Visual Administrator)**
 - Log and trace files for the runtime environment and the running applications are automatically registered
 - Define the properties, such as: activating log monitoring, etc..
- **As a central Log Viewer** to centrally display log and trace files
 - consisting of: Log Viewer and an additional remote server
 - You can display log files for a system landscape centrally in the central Log Viewer client if a remote server is running on every host.
- **Command Line Log Viewer**
 - Delivered with the Standalone Log Viewer (lv.bat)
 - Displays only local log files
 - Can be activated during the deployment of applications
 - Converts binary data to a readable format



Note: This lesson focuses on the integrated Log Viewer in NWA and the central Log Viewer.

The Log Viewer in the SAP NetWeaver Administrator

The Log Viewer runs as a service in SAP NetWeaver AS Java. As soon as the SAP Logging API is aware of a new log, the log is automatically included and can be displayed in the NWA's Log Viewer or in the integrated Log Viewer.

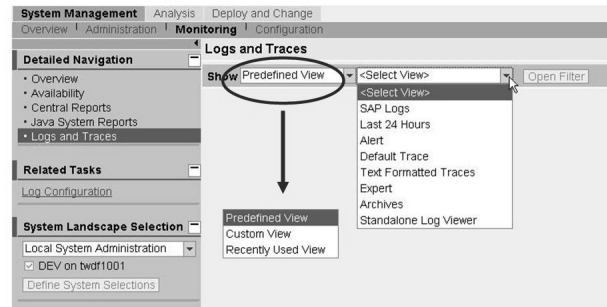


Figure 173: Log Viewer in the NWA: Predefined Views

The log and trace files are automatically registered when SAP NetWeaver AS Java is started so that they can be displayed using the above-mentioned. Log Viewer variants. This is achieved by means of special XML connection files which are written to the *tmp* (SAP NetWeaver AS Java installation directory at start time). In the NWA, you can call the Log Viewer via the following path *System Management* → *Monitoring* → *Logs and Traces*. Multiple predefined views are available (figure: Log Viewer in the NWA: Predefined Views) and you can also save your own user-defined views. The predefined views do not usually display all the log and trace entries. Instead these are restricted by filters in the views themselves.



- **Last 24 Hours**
Shows log and trace entries for the last 24 hours
- **SAP Logs**
Shows log entries but no trace entries
- **Alert**
Shows log entries with severity level “Error” or “Fatal”
- **Default Trace**
Shows trace entries but no log entries
- **Expert**
Shows all log and trace entries without restriction
- **Text Formatted Traces**
Shows file contents which are not of type “ListLog”

You can use the *Open Filter* button to activate further restrictions to the selected view by means of filters and save this as a user-defined (Custom) view. For more information, see the figure “Log Viewer in the NWA: Filters”.

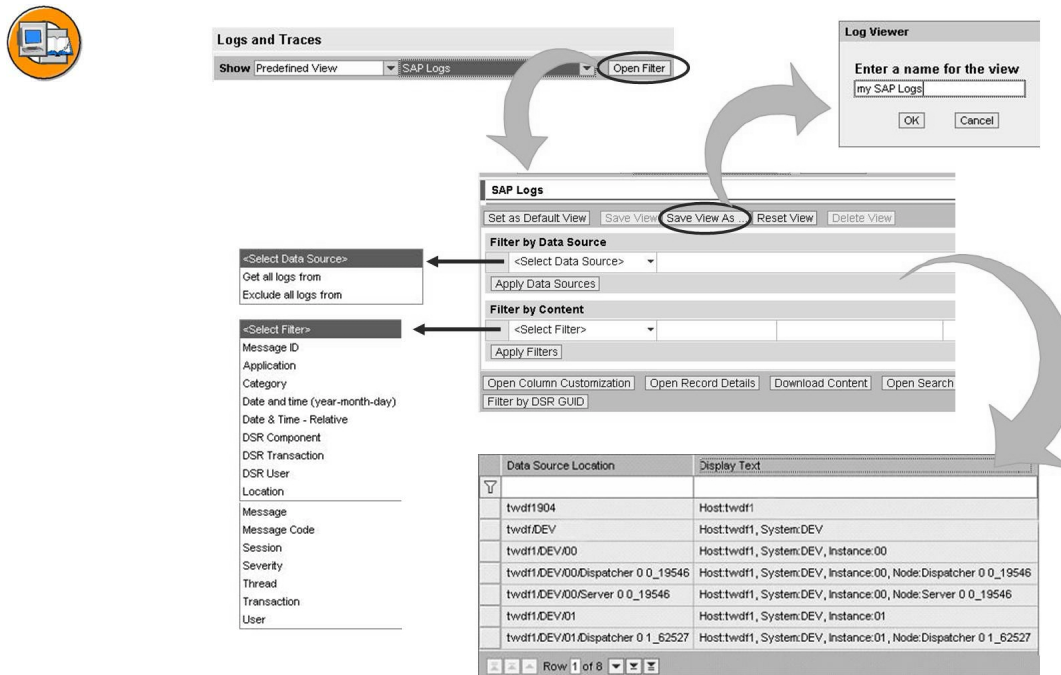


Figure 174: Log Viewer in the NWA: Filters

If you set the filter to “Data Source” then you can restrict the view to different systems, instances or individual nodes. The “Content” filter can be used to filter the data specified in the screen. Data of interest here may be, for example, Message, Date, User, DSR Transaction, Category, Location. The predefined view which has been fine-tuned in this way can then be stored as a user-defined view. If you identify an entry for which you want to see the associated messages (possibly from other files or related log and trace information) then filtering for the “DSR Transaction” is often useful. Figure “Log Viewer in the NWA: Column Customization” illustrates how you specify which columns are displayed in the NWA Log Viewer.



Caution: The new “Custom” view created in this way contains restrictions to the original view which are not directly visible. However, this can be identified via the *paper clip* button (see figure “Log Viewer in the NWA: Column Customization”). It is therefore more practical to create your own views from the Expert view which does not contain any “invisible” restrictions.



Open Column Customization Open Record Details Download Content Open Search **Records to Be Displayed** 10 Filter by DSR GUID

Columns

Move Up Move Down

- ☒ Date
- ☒ Time
- ☒ Message
- ☒ Category
- ☒ Location
- ☒ Application
- ☐ Thread
- ☐ Data Source
- ☐ Message ID
- ☐ Argument Objects
- ☐ Arguments
- ☐ DSR Component
- ☐ DSR Transaction
- ☐ DSR User
- ☐ Message Code
- ☐ Session
- ☐ Transaction
- ☐ User
- ☒ Host
- ☐ System
- ☐ Instance
- ☒ Node

Apply to All Views

Details	Severity	Date	Time	Message	Category	Location
▶	i info	2007-08-15	13:26:44:387	Service jmx_notification started. (110 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:44:293	Service userstore started. (0 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:44:262	Service classpath_resolver started. (32 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:44:215	Service p4 started. (953 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:43:199	Service timeout started. (187 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:43:199	Service trex.service started. (78 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:43:027	Service runtimeinfo started. (15 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:42:980	Service file started. (62 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:42:980	Service cross started. (15 ms).	/System/Server	com.sap.engine.core.service630.containe
▶	i info	2007-08-15	13:26:42:918	Service memory started. (0 ms).	/System/Server	com.sap.engine.core.service630.containe

Figure 175: Log Viewer in the NWA: Column Customization

Alongside the columns selected in the figure “Log Viewer in the NWA: Column Customization”, you should also select the columns “User” and “Data Source”. This indicates the data source for the entry. Here, you can also see whether you are viewing log or trace information and the system, instance or node for which the entry was made.

You use the *Open Record Details* button to display additional information for an entry. You can use the “Open Search” button to search in the filtered entries. The buttons between the up and down scroll buttons are extremely interesting. They can be used to display entries which were written before or after the selected entry but which were hidden due to the filter option. *Filter by DSR GUID* acts in a similar way to a manual filter on a DSR transaction. However, it temporarily deactivates all content filters. If you press the *Filter by DSR GUID* button again, all the original content filters are restored. If you choose the “Go to Newest Records” button or update the filters, the display of the entries is refreshed.



Hint: The *Download Content* button is very useful. This makes it possible to store the filtered data in a .csv file on the front-end. This file consists of a comma-separated list which you can scroll through conveniently in Microsoft Excel and find interesting messages quickly.

In Microsoft Excel, you can import the data via *Data* → *Import External Data* → *Import Data*. As the original file type, you should specify “Delimited” and not “Fixed Width”. The delimiting character is the “comma”.

In the Expert view, there is, alongside the already familiar filters, one more, the “Log Browser” with which you can create restrictions yourself in the predefined views. This has the advantage that if a filter is open, you can always see immediately which data is selected or hidden.

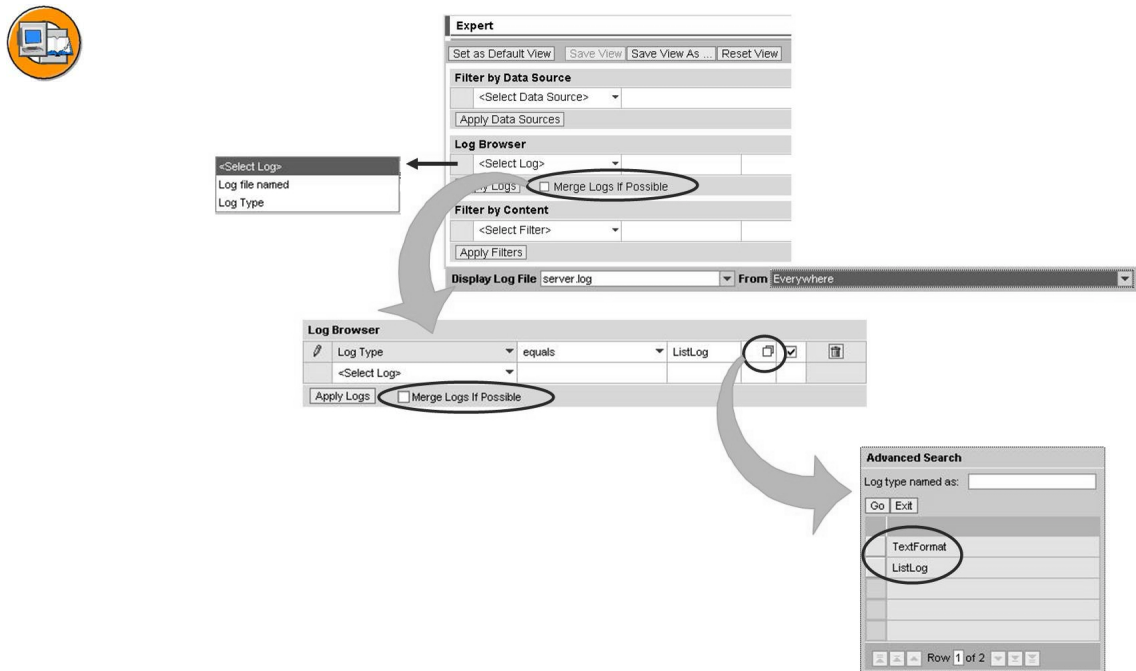


Figure 176: Log Viewer in the NWA: Expert View

The figure “Log Viewer in the NWA: Expert View” shows that the Log Browser provides two different selections. You can use “Log Type” to choose between ListLog and TextFormat. If you choose the ListLog restriction, then both trace and log data is available for display. This data is stored in different files as already discussed at the start of the lesson (see also figure: “ListLogs in the File System”). You can use *Display Log File* to select a file whose data is to be displayed. You can select “From” to restrict the selection to a specific node or choose “Everywhere” to display the combined data from all nodes. If, as in the predefined views, you want to display the combined data from all the ListLogs then you should select **Merge Logs if Possible**. If you select “Log File named” then you can specify explicitly the files from which data is to be included or excluded.. Thus, “Log Type equals ListLog” together with “Log

file named as DefaultTrace*” and “Merge Logs if Possible” yields the same result as the predefined Default Trace view. If you want to see the data as in the SAP Logs view, you should instead simply choose “Log file named different from DefaultTrace*”.

Files in text format cannot be combined using “Merge Logs if Possible”. If you choose TextFormat, then you can, for example, also display files such as the dev_dispatcher file.



Hint: If in the Log Browser you only select “Merge Logs if Possible” and do not specify any further restrictions then you can use “Display Log File” to select a combination of all the log and trace files or the individual text format files.

The Integrated Log Viewer

The Log Viewer runs as a service in SAP NetWeaver AS Java. As soon as the SAP Logging API is aware of a new log, the log is automatically registered and can be displayed in the NWA's Log Viewer or in the integrated Log Viewer. If the SAP NetWeaver AS Java is not available, you cannot display any log files with the integrated Log Viewer. The integrated Log Viewer is available in the Visual Administrator.

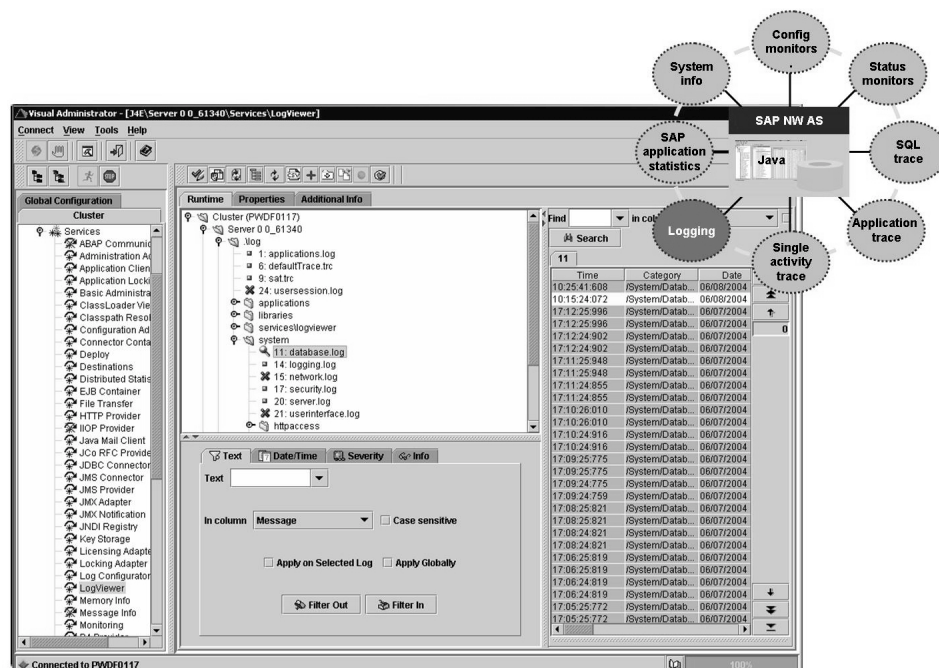


Figure 177: Integrated Log Viewer

The integrated Log Viewer is primarily used for displaying or registering log files. Registration is performed either automatically or manually. As soon as the Log Viewer service starts, the log information is automatically evaluated. The SAP NetWeaver AS Java has its own XML files, which are imported and therefore automatically provide log/trace information. If the Log Viewer does not automatically enter files, you can add them manually.

For **manual** registration of log files, open the Visual Administrator and choose the *Runtime* tab page in the Log Viewer service, and then the *add file* icon. It is a prerequisite that access to the directories is permitted. You can find the *Logviewer_MonitorablePath* parameter in a tab in the Log Viewer service (Visual Administrator). There you enter the directories that you want to be able to select in the Log Viewer for the *Manual Registration of Log Files* function.

The Central Log Viewer

The central Log Viewer provides access to log files even if the corresponding SAP NetWeaver AS Java is no longer running. The central Log Viewer can display ASCII-based logs of other applications, such as a database.

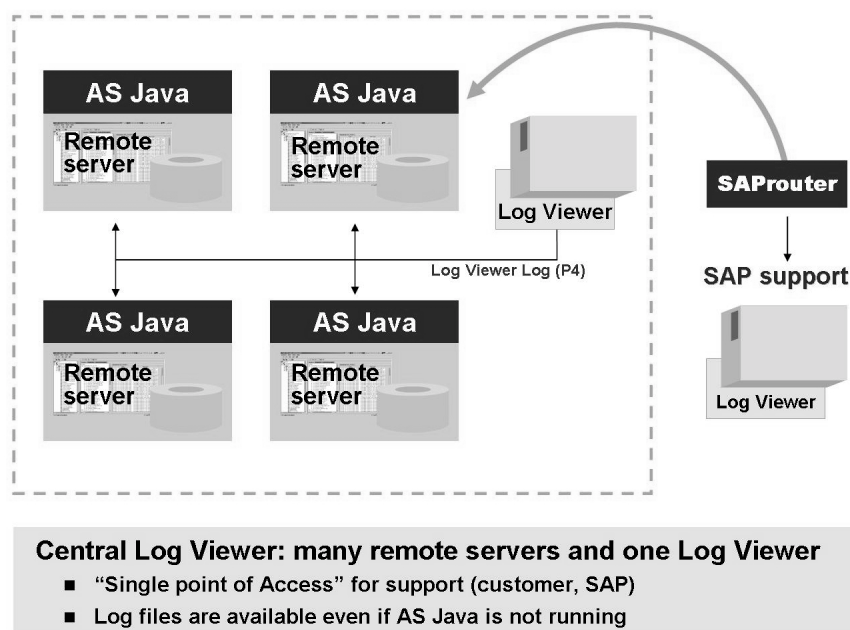


Figure 178: Central Log Viewer 1/3

The central Log Viewer consists of the Log Viewer and an additional remote server. The central Log Viewer can connect to the SAP NetWeaver AS Java directly with the P4 port. A remote server does not need to be started for this but the SAP NetWeaver

AS Java must be running. Generally it is more useful to connect to the remote server that is started on a remote SAP NetWeaver AS Java. Port 26000 is used for this. This connection also works without problems when the remote SAP NetWeaver AS is unavailable. When you start the central Log Viewer, you see the following screen:

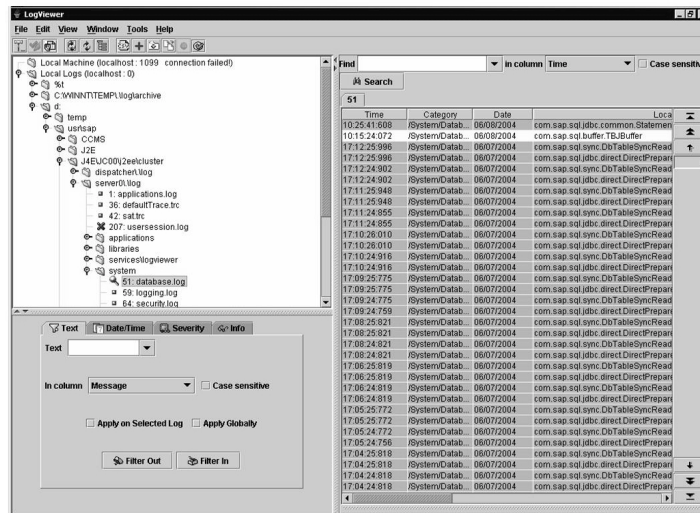


Figure 179: Central Log Viewer 2/3

After the installation of SAP NetWeaver AS Java, the Log Viewer exists automatically and is ready for use. The scripts of the central Log Viewer are in the directory *admin/logviewer_standalone* and contain a Log Viewer and a remote server start script. You can define the remote server as a service under Microsoft Windows and as a daemon under UNIX. In the local Visual Administrator, you can call the *LogViewer* service and check whether all registered log files are visible. The script *logviewer.bat/logviewer.sh* is started on the host on which the logs are to be displayed in the central Log Viewer. The remote server is started by a script, *remoteserver.bat/remoteserver.sh*, on all Java instances that are to be monitored with the central Log Viewer. In the case of Microsoft Windows, it can also be started from the subdirectory *server*. You need to configure a connection from the Log Viewer to the servers. You can create a connection to the server by choosing *File → Connection to Server*.

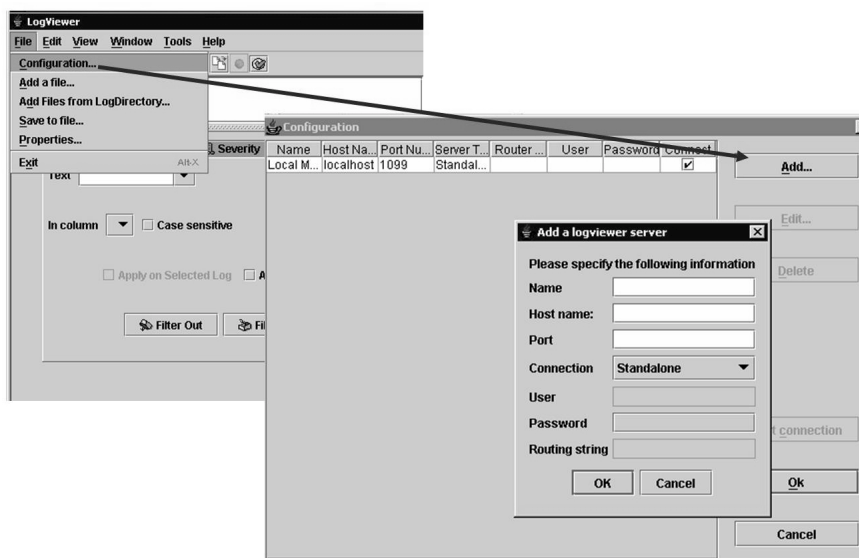


Figure 180: Central Log Viewer 3/3

After you have connected to the remote server you can register log files. The remote server only allows access to the directories that are defined in the *Logviewer_MonitorablePath* parameter in the file *logviewerServer.properties* (directory `<j2ee root directory>\admin\logviewer_standalone\server`; such as `G:\sap\<SID>\<instance>\j2ee\admin\logviewer_standalone\server`). You can change the *Logviewer_MonitorablePath* parameter. The new parameter values are active after restarting the remote server.

Alongside the central Log Viewer, the NWA is also able to display data for a remote server. This is possible using the “Standalone Log Viewer” view.

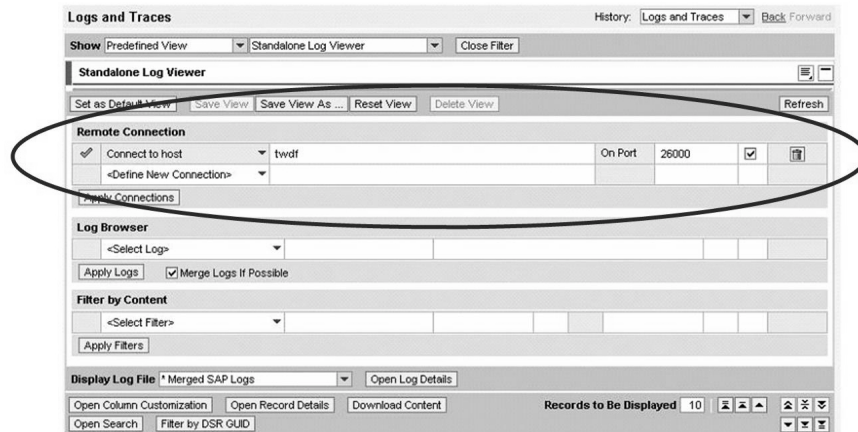


Figure 181: Log Viewer in the NWA: Standalone

The figure “Log Viewer in the NWA: Standalone” presents the available filter possibilities. This view can also be stored as a user-defined Custom view. It is of course also possible to read data from several different remote servers and combine their ListLogs by choosing “Merge Logs if Possible”. For example, it is possible to save individual views for different SAP systems or save views in which data from multiple systems is displayed.

Logging and Tracing

Log files are displayed in the Log Viewer. There are two types of log files: files for logging, and files for tracing.

Logging means:



- Recording normal and exceptional events
- Runtime information of a system or an application is written to log files
- Active during normal operation
- Logs are structured into **categories**, which are logical areas/topics. Predefined categories are:
 - System (Server, Network, Database, Security)
 - Application
 - performance
- Each category points to one or more log destinations (storage locations in the file system)

Tracing means:

- Recording the process flow of an application
- Use during development and for error detection in the production environment
- All traces are stored in the default.x.trc files
- Traces are structured into **locations**.



Note: Locations represent defined coding areas such as classes or software packages.

The traces and logs are displayed in the logging/tracing infrastructure. The logging/tracing infrastructure for SAP NetWeaver AS Java consists of:



- consisting of: SAP Logging API, Log Manager, Log Controller
- is configured via: Log Configurator service
- Is displayed in: Log Viewer

SAP Logging API, Log Manager, Log Controller

The **SAP Logging Infrastructure** consists of the SAP Logging API, the Log Manager, and the Log Controller. The **Log Manager** reads the configuration files and informs the Log Controller of the severity with which log information is being written for an application in a log file. The **Log Controller** makes it possible to store data in a file, console, or another output target. In the Visual Administrator, all objects for which log or trace files can be written are displayed as Log Controllers.

The Log Manager is a central manager in the structure of a J2EE server. This manager is the first manager that is started. The storage location for all traces is configured here. There is only one directory and one file per node in which all traces are written.

All log files are written to the directory *J2EE Root/cluster/<server or dispatcher>/log* (such as */usr/sap/<SID>/<instance>/j2ee/cluster/dispatcher/log*).

The log files can have different **severities**. This means that in a log file, for example, only errors, only errors and warnings, or all information in debug mode is written. The following severities exist:



- ALL (Low)
- DEBUG
- PATH
- INFO
- WARNING
- ERROR
- FATAL (High)

The severities ALL/DEBUG and PATH are intended for trace files. All messages with the severity INFO, WARNING, ERROR, or FATAL should be written as entries in a log file.

Configuration of Logs and Traces in the Log Configurator Service

The **Log Configurator service** provides a runtime environment for configuring logs and traces. In the Visual Administrator, you can use the Log Configurator service to perform logging/tracing configuration for components of SAP NetWeaver AS Java and deployed applications. Log configuration can also be performed with the NWA.

Each application provides a file, *log-configuration.xml* that contains the initial configuration of the logging and tracing for this application. The SAP NetWeaver AS Java has its own XML file in the *tmp* directory (in the root directory of the installation), which is imported and therefore automatically provides log/trace information.

The Visual Administrator offers two configuration modes: *simple* and *advanced*. You can maintain *log destinations*, *log formatters*, and filters only in the advanced mode. The Log Configurator service is in the Visual Administrator. Navigate as described in the following figure:

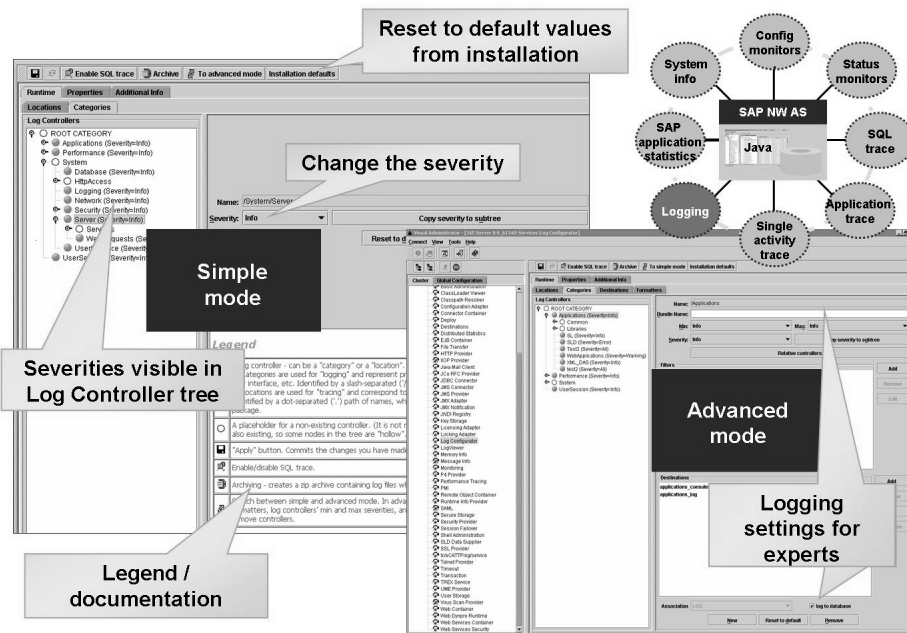


Figure 182: Log Configurator service

You can perform the following actions in the Visual Administrator using the Log Configurator service:



- Change the severity
- Add, change, and delete log destinations (storage locations)
- Add, change, and delete log formatters
- Add, change, and delete log controllers
- Archive log files



Hint: You usually only need to change the severities. All other settings are intended for experts.

Log formatters are formatters for files in different formats such as XML, trace, and list format.

You can configure log destinations for categories (log files) and locations (trace files). A **log destination** allows you to determine where the log/trace files are stored.

All objects for which log or trace files can be written are shown as **log controllers** in the *Log Configurator service*.

Adjusting Log Destinations

In the Visual Administrator, choose the service *Log Configuration* under Server. In advanced mode, choose the *Log Destinations* tab page. You can create new log destinations or changing existing destinations there. You make settings for the storage locations are made in the *Pattern*. You can also maintain the log formats (field *Log Formatter*) and filter settings here.



Note: If you are creating a new log destination, you should define the file type. There are two file types, *FileLog*, and *ConsoleLog*. In the case of *FileLog* type, it is also necessary to make the following specifications: *Pattern*, *Maximum File Size* and *Number of files*.

You can see the assignments for the log destinations in advanced mode on the *Categories* and *Locations* tab page.

You usually only need to adjust log destinations if, for example, you are working with the UNIX operating system and want to view log files on the console. In this case, you need to change the log format to *ConsoleLog*. Log formatters are directly connected to *LogDestinations*. If you want to change the log format for a log destination, you can do this only in advanced mode using the *Log Destination* tab page.

Adjusting Log Formatters

You can see the *Log Formatters* tab page only in advanced mode, and can change existing log formatters there. You need to maintain the fields *Pattern* and *Type*. SAP delivers the *Types* *ListFormatter*, *TraceFormatter*, and *XMLFormatter*. *ListFormatter* means that the log entry can be processed by an application such as the Log Viewer. *XMLFormatter* outputs an element in the XML style. *TraceFormatter* is a formatter that can be read by users. Only with *TraceFormatter* can you maintain the second field *Pattern*.



Hint: It is not usually necessary to maintain log formatters, since SAP delivers the appropriate log formatters.

Changing Severities

You can also change the severities for log controllers and severities for log destinations in the *Log Configurator service*. The severity settings for log controllers have a higher priority than those for log destinations. If, as in the figure, the severity *Info* is entered for the log controller, you can only meaningfully choose the same severity or a higher severity (such as *Warning*, *Error*, and so on) for the log destinations.

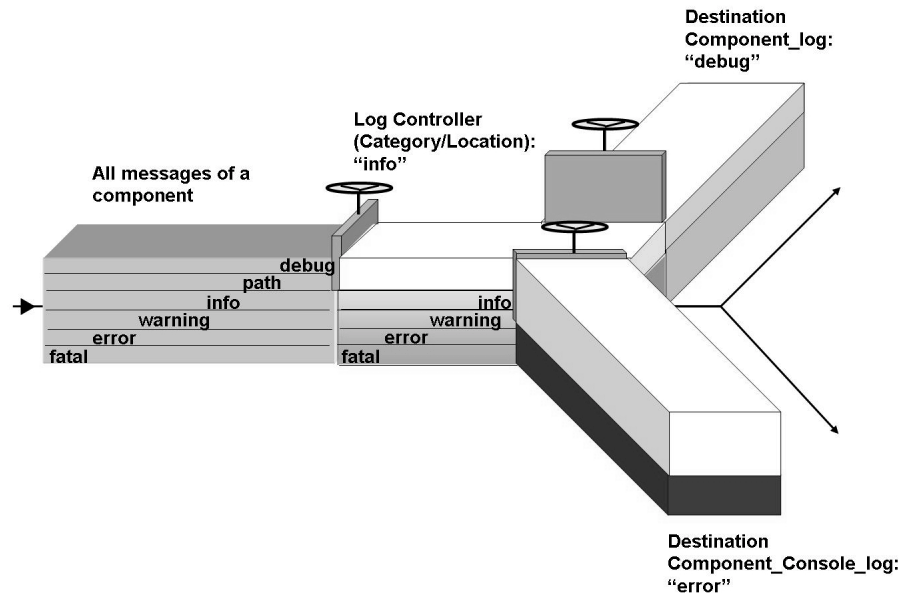


Figure 183: Logging API Logic

In the Visual Administrator, choose the service *Log Configuration* under *Server*. Select a log controller from the tab pages *Categories* or *Locations* and change the *Severity* field.

log archiving

The Log Configurator service provides the **Log Archiving** option. Log files are automatically archived at specific intervals. You can activate this function using the Visual Administrator (*Server* → *Services* → *Log Configurator* → *Properties*). Change the parameter *ArchiveOldLogFiles* to the value *ON*. By default, the archives are written on the SAP NetWeaver AS in the directory *<J2EE root directory>/<server or dispatcher>/log/archive* (such as */usr/sap/<SID>/<instance>/j2ee/log/archive*). The parameter *ArchivesDirectory* defines the storage location of the archives. The archives themselves are not automatically deleted. You need to do this manually.

Configurations with the Log Manager

To call the Log Manager, choose the following path in the Visual Administrator: *Server or Dispatcher* → *Kernel* → *Log Manager*.

Configure the Storage Location for Trace Files

The Log Manager offers the option to specify the storage location for all configured traces and to specify a separate directory for individual traces. By default, all traces are written to one file in a specific directory. You can find them under: *<J2EE root directory>\cluster\server0\log\defaultTrace.trc*, e.g. *usr\sap\<SID>\<instance>\cluster\server0\log\defaultTrace.trc*.



All traces are written in one file:

`usr\sap\<SID>\<instance>\j2ee\cluster\server0\log\defaultTrace.trc`

Global Configuration		Properties	
Cluster		Key	Value
<ul style="list-style-type: none"> * Dispatcher Server <ul style="list-style-type: none"> Kernel ClassLoaderManager ClusterManager ConfigurationManager ConnectionsManipulator IpVerificationManager LicensingManager LockingManager LogManager PoolManager PortsManager 		ConsoleLogs_UseSapAPI	YES
		DatabaseLogs_AttemptsTimeout	5
		DatabaseLogs_DaysToKeep	7
		DatabaseLogs_Enabled	NO
		DatabaseLogs_InitAttempts	20
		DatabaseLogs_Severity	NONE
		DefaultTraceFile_Count	20
		DefaultTraceFile_Limit	10485760
		DefaultTraceFile_Pattern	/log/defaultTrace.trc
		ForceSingleTraceFile	YES
		SQLTraceInitiallyEnabled	NO
		SingleTraceFile_UnrestrictedLocations	

Specify a location for a separate trace file (location name → Log Configurator Service)



You must update and restart the server



Figure 184: Default Trace

Modifying Severities with the SAP NetWeaver Administrator

With SAP NetWeaver Administrator, you can make the settings for the log and trace severities in the same way as with the Visual Administrator. In the figure “Log and Trace Configuration in the NWA”, you can see the path for calling the log configuration. There is a separate view for both the categories and the locations and the severities can be adapted in these views.

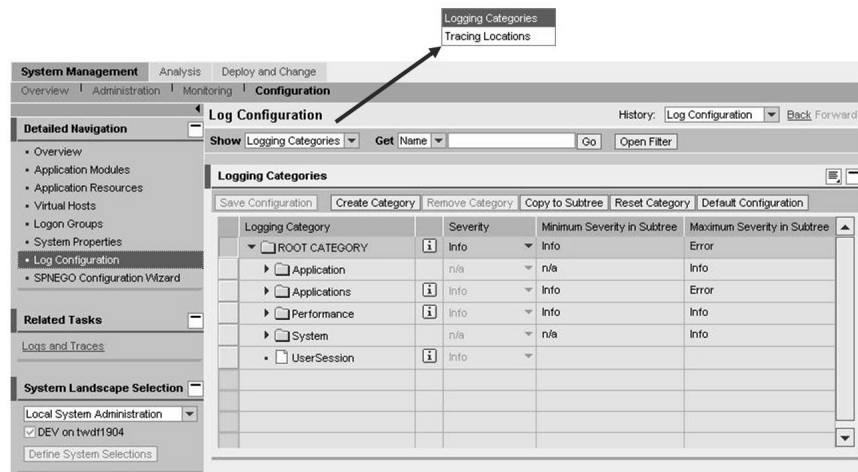


Figure 185: Log and Trace Configuration in the NWA

As you can see in the figure “Severities in the NWA”, it is possible to set the severity for all the nodes of a category or location system-wide in the top area.

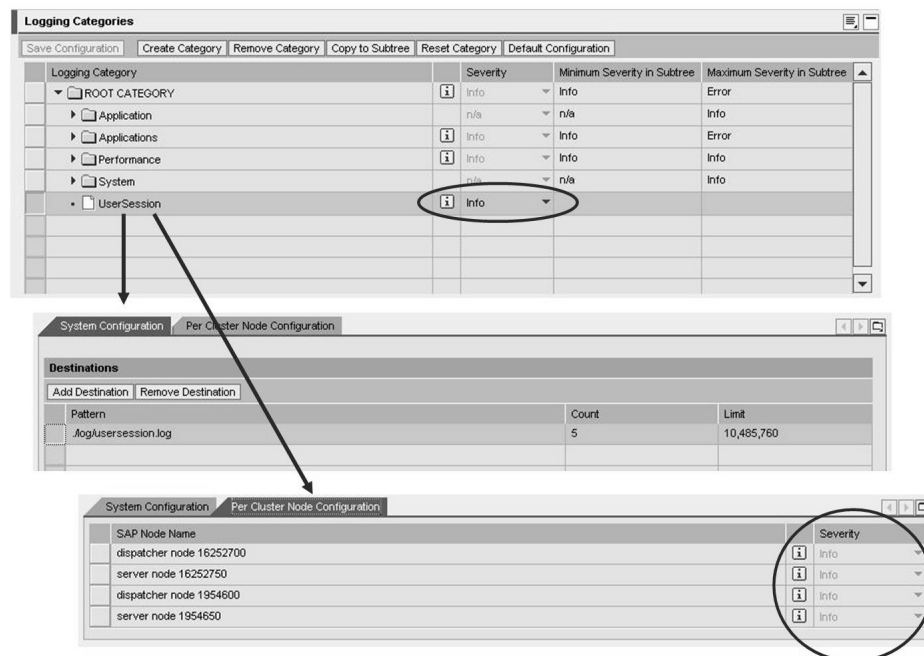


Figure 186: Severities in the NWA

In the lower Log Configuration area, you can switch between the “System Configuration” and “Per Cluster Node Configuration” tabs (figure: “Severities in the NWA”). In the System Configuration, you see the storage location defined under Log Destination and the name of the file to which the entries are written. You can use the “Per Cluster Node Configuration” view to set other severities for individual nodes. If you want to reset a category or location to the value shipped by SAP then you can do this using the *Reset Category* or *Reset Location* button respectively.

Exercise 21: Log Viewer and Log Configuration

Exercise Objectives

After completing this exercise, you will be able to:

- Change the severity in the Log Configurator service
- View log files in the integrated Log Viewer
- Operate the central Log Viewer

Business Example

You are working with SAP NetWeaver AS Java and want to know more about the options for configuring and evaluating log files. Since a great deal of log information is created in the SAP NetWeaver AS Java environment, it is important to be familiar with a tool that automatically displays the log files for stable operation.

Task 1: User-Defined Views in the NWA Log Viewer

Create your own view in the NWA Log Viewer which provides you with information about when the nodes in your system were started. To do this, use the ListLogs of the predefined Expert view. Note: The first manager to be started is the LogManager.

1. Log on to your system's NWA and switch to the display of logs and traces.
2. Select the Expert view and create your own filter which provides you with information about the start of the LogManager. Save these settings as a user-defined view.
3. Add the columns "Data Source, DSR Transaction, User, System, Instance" to this view.

Task 2: Troubleshooting with the Log Viewer in the NWA

Scenario: Your SAP NetWeaver AS Java does not report any data to the SLD. Use the Log Viewer in the NWA to find out what the problem could be.

1. Create a new view. To do this, use the view created in the previous task as a template and name it, for example, my Expert SLD.
2. Create a filter which supplies all the entries in which SLD occurs in the Message column.
3. Search for a message that will help you identify the problem.

Continued on next page

4. In the above message, identify a log or trace message and determine which node it has been sent by.

Task 3: Correcting the SLD Problem

1. Log on to the Visual Administrator.



Hint: If you are still logged on to the Visual Administrator, close the Visual Administrator and then restart it.

2. Choose *Server* → *Services* to go to *SLD Data Supplier*
3. Here, choose the *blue lightning button* to trigger data transfer to the SLD. You will see an error message. If, in the NWA, you now perform a refresh in your *my Expert* SLD view, you will now see a new error message. You are therefore on the right track to correct the problem.
4. Now go back to the Visual Administrator, choose the *SLD Data Supplier* service and go to *HTTP Settings*. Here, enter the data communicated to you by your instructor under *Host, Port, User and Password*. Save your input.
5. Choose the *blue lightning button* again to trigger data transfer. You see the message “Data transfer performed successfully”. Your SAP NetWeaver AS Java system information is now visible in the SLD.

Result

Congratulations! You should now find no further *error* messages concerning SLD in your system.

Task 4: Log Configuration in the NWA

In the previous task, you saw that the *SLD Data Supplier* had a problem. You should now adapt the set *Severity* level.

1. In the NWA, go to Log Configuration
2. Choose the appropriate view.
3. Search for the appropriate entry for the SLD service.
4. Change *Severity* system-wide from *error* to *warning* and save your input.
5. Log off the NWA.

Continued on next page

6. Log on to the NWA again. You will see a message that the SLD is not accessible even though we have just set up the connection. Only local administration is possible at present.
7. In the Logs and Traces, call up your *my Expert* SLD view again and examine the new messages

Task 5: OPTIONAL: Log Configurator service

Change a severity so that less information is written.

1. Log on to the Visual Administrator.
2. Set the severity for the Database category to *Error*.

Task 6: OPTIONAL: Log Viewer

Working with Log Viewer.

1. Open the integrated Log Viewer and view the database log files for the last few days. View the detailed information for the last message that was modified.
2. Start the central Log Viewer. Search for messages with the severity *Warning*.
3. OPTIONAL: Enter the following log file in the central Log Viewer: *G:\usr\sap\<SID>\<instance>\j2ee\admin\logviewer-standalone\Service_Readme.txt*

Solution 21: Log Viewer and Log Configuration

Task 1: User-Defined Views in the NWA Log Viewer

Create your own view in the NWA Log Viewer which provides you with information about when the nodes in your system were started. To do this, use the ListLogs of the predefined Expert view. Note: The first manager to be started is the LogManager.

1. Log on to your system's NWA and switch to the display of logs and traces.
 - a) In the browser, start the URL <http://<hostname>:<port>/nwa> .
 - b) Navigate to *Monitoring* → *Logs and Traces*.
2. Select the Expert view and create your own filter which provides you with information about the start of the LogManager. Save these settings as a user-defined view.
 - a) Select the predefined “Expert” view.
 - b) Open the Filter view.
 - c) In the “Log Browser”, select the “Log Type”. If “equals ListLog” has not yet been selected, make sure that it is entered. Select “Merge Logs if Possible”.
 - d) Under “Filter by Content”, select the Message filter and filter on “contains” “LogManager started”.
 - e) Choose the *Refresh* button.
 - f) Save this view under a descriptive name, e.g. my Expert LogMgr Start.
3. Add the columns “Data Source, DSR Transaction, User, System, Instance” to this view.
 - a) Select *Open Column Customization*.
 - b) Select the above-mentioned columns in addition to those that are already selected.
 - c) Save your view.

Continued on next page

Task 2: Troubleshooting with the Log Viewer in the NWA

Scenario: Your SAP NetWeaver AS Java does not report any data to the SLD. Use the Log Viewer in the NWA to find out what the problem could be.

1. Create a new view. To do this, use the view created in the previous task as a template and name it, for example, my Expert SLD.
 - a) In the NWA, switch to your view from the previous task.
 - b) Create a new view by choosing the button *Save View as ...*
2. Create a filter which supplies all the entries in which SLD occurs in the Message column.
 - a) Modify the Content Filter to search for SLD instead of “LogManager started”.
 - b) Choose the “Apply Filter” button to apply the modified filter.
 - c) Save your settings.
3. Search for a message that will help you identify the problem.
 - a) You should now see an “error” message which, at first glance, does not seem to indicate any link with the SLD. Choose *Open Record Details* to examine the message in more detail.
 - b) Here, you will find information indicating that the message has something to do with **sldserv** HTTP communication with the bridge of the destination SLD. The start of the message tells you that communication to the host “twdfnowhere:50000” did not function correctly and that the URL may be incorrect.

This is the problem: Your SAP NetWeaver AS Java is trying to transfer data for the SLD to the wrong host with an incorrect port.
4. In the above message, identify a log or trace message and determine which node it has been sent by.
 - a) In the Data Source column, you can see that this is a trace message since it was written to the defaultTrace. Here, you can also see that it was reported by a server process.

Continued on next page

Task 3: Correcting the SLD Problem

1. Log on to the Visual Administrator.



Hint: If you are still logged on to the Visual Administrator, close the Visual Administrator and then restart it.

- a) Start *go.bat* from the directory *g:\sap\<SID>\<central instance>\j2ee\admin*
 - b) Log on with the appropriate user and password. Your instructor will give you the relevant information.
2. Choose *Server* → *Services* to go to *SLD Data Supplier*
 3. Here, choose the *blue lightning button* to trigger data transfer to the SLD. You will see an error message. If, in the NWA, you now perform a refresh in your *my Expert* SLD view, you will now see a new error message. You are therefore on the right track to correct the problem.
 - a) You perform the refresh by clicking the *Refresh* button or the button that allows you to display the most recent entries (*Go to newest Records*)
 4. Now go back to the Visual Administrator, choose the *SLD Data Supplier* service and go to *HTTP Settings*. Here, enter the data communicated to you by your instructor under *Host, Port, User and Password*. Save your input.
 5. Choose the *blue lightning button* again to trigger data transfer. You see the message “Data transfer performed successfully”. Your SAP NetWeaver AS Java system information is now visible in the SLD.

Result

Congratulations! You should now find no further *error* messages concerning SLD in your system.

Task 4: Log Configuration in the NWA

In the previous task, you saw that the *SLD Data Supplier* had a problem. You should now adapt the set *Severity* level.

1. In the NWA, go to Log Configuration
 - a) Go to *System Management* → *Configuration* → *Log Configuration*

Continued on next page

2. Choose the appropriate view.
 - a) In the previous task “Troubleshooting with the Log Viewer in the NWA”, we saw that we are dealing with trace information and we therefore choose *Tracing Locations*
3. Search for the appropriate entry for the SLD service.
 - a) In the search field, enter “sld”. Here, you will see a number of entries. *com.sap.sldserv* is the right one here as you already know from the error message. Select it.
4. Change *Severity* system-wide from *error* to *warning* and save your input.
 - a) You may have to scroll down a little in the top window. *sldserv* should already be selected. In the top window, click on *Severity* and select *Warning*. Choose *Save Configuration* to save the new severity level.
5. Log off the NWA.
 - a) To do this, select the *Log Off* button at the top right.
6. Log on to the NWA again. You will see a message that the SLD is not accessible even though we have just set up the connection. Only local administration is possible at present.
7. In the Logs and Traces, call up your *my Expert* SLD view again and examine the new messages
 - a) Here you will see a warning relating to insufficient rights. **As far as access to the SLD is concerned, everything is OK.** You would not have seen this message if you had not adapted the severity level. The message tells us that the rights to read data concerning the systems registered with the SLD are still missing. We will examine this issue if necessary in the appendix to the NWA if the NWA is set up centrally.

Task 5: OPTIONAL: Log Configurator service

Change a severity so that less information is written.

1. Log on to the Visual Administrator.
 - a) Start *go.bat* from the directory *g:\sap\<SID>\<central instance>\j2ee\admin*
 - b) Log on with the appropriate user and password. Your instructor will give you the relevant information.

Continued on next page


2. Set the severity for the Database category to *Error*.
 - a) Select the appropriate server and open the Services there.
 - b) Open the *Log Configurator* service, and choose the *runtime* tab page.
 - c) On the *Categories* tab page, search for the log controller *System* → *Database*.
 - d) You can now set the severity to *Warning* or *Error* in the window on the right. Save your entries by choosing the *Apply* icon and selecting *Apply to all server nodes*.

Task 6: OPTIONAL: Log Viewer

Working with Log Viewer.

1. Open the integrated Log Viewer and view the database log files for the last few days. View the detailed information for the last message that was modified.
 - a) Start *go.bat* from the directory *g:\usr\sap\<SID>\<instance>\j2ee\admin* and log on with the appropriate user and password. Your instructor will give you the relevant information.
 - b) Select the appropriate server and open the *LogViewer* service there. Navigate to the database log files by selecting the *runtime* tab page, and branching to *Server* → *G:\usr\sap* → *<SID>\<instance>* → *j2ee\cluster\server* → *log* → *system* → *database.log*. You can view the database log data by double-clicking it. Select the *Date/Time* column, and sort it by double-clicking.
 - c) Switch to the table view (icon: *View logs as table/tree*) and sort on the *Last modified* column. To display more information about the message with the newest date, double-click the entry.
2. Start the central Log Viewer. Search for messages with the severity *Warning*.
 - a) Start *logviewer.bat* from the directory *g:\usr\sap\<SID>\<instance>\j2ee\admin\logviewer-standalone*.
 - b) Navigate to *G:\usr\sap* → *<SID>* → *<instance>* → *j2ee* → *cluster* → *server* → *log* → *defaulttrace.trc* under *localhost*.
 - c) Now use the filter to select messages with the severity *Warning*. Use the icon *Include in current Logfile*. The system displays only the messages with the severity *Warning*.

Continued on next page

3. OPTIONAL: Enter the following log file in the central Log Viewer: *G:\usr\sap\<SID>\<instance>\j2ee\admin\logviewer-standalone\Service_Readme.txt*
 - a) Start the central Log Viewer as described in task 2, and connect to the *localhost* connection.
 - b) Choose the *Add File* icon, navigate to the specified file, and add this by choosing *Add*.
 **Note:** Pay attention to the position of the cursor when doing so, since the log file is added at this location.
 - c) Now check that the file is visible in the central Log Viewer.



Lesson Summary

You should now be able to:

- Operate the integrated and the central Log Viewer
- Explain the difference between logging and tracing
- Discuss the most important functions of the Log Configurator service
- Use the Log Configurator service to adjust the severity of log files

Lesson: Availability Monitoring

Lesson Overview

SAP makes available availability monitoring using the *Generic Request and Message Generator* (GRMG). You can use it to monitor both technical components of SAP NetWeaver AS Java and entire Java applications. You can use this availability monitoring with only a few configuration steps.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe how an availability check using the GRMG works technically
- Configure an availability check
- Explain which steps a developer must perform to create a GRMG-compatible application

Business Example

You are using SAP NetWeaver AS Java and want to be notified as quickly as possible if a Java application or technical component of an SAP Web AS Java is not running. In this case, it is useful to configure an availability check using the GRMG.

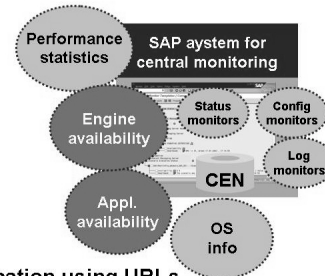
Fundamentals of Availability Monitoring

SAP provides the tools for monitoring the SAP NetWeaver AS Java and Java applications. This availability monitoring is based on the *Generic Request and Message Generator* (GRMG). You can use the GRMG to monitor the availability of technical components and the availability of entire business processes.



GRMG: Generic Request and Message Generator

Central infrastructure for availability monitoring of Java-based components and applications



Functionality:

- GRMG infrastructure periodically calls GRMG application using URLs
- GRMG request (XML) is sent to the GRMG application to check availability
- GRMG application returns a response (XML)

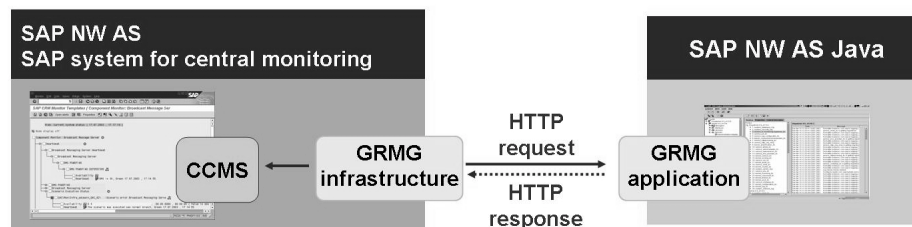


Figure 187: Availability Monitoring

The GRMG consists of two parts, both of which are required for a functioning GRMG environment:

- **GRMG infrastructure**

The GRMG infrastructure is part of the monitoring architecture of the Computing Center Management System (CCMS) of an SAP NetWeaver AS ABAP. Its task is to send a request (the GRMG request) to the GRMG application, to receive its response (the GRMG response), and to display this response in the CCMS Alert Monitor.

- **GRMG application**

The GRMG application performs the actual availability monitoring. From a technical point of view, it is a Java Server Page (JSP), a servlet, or a Business Server Page in an SAP NetWeaver Application Server with a defined interface that is called by the GRMG infrastructure. The GRMG request and GRMG response are messages in a special XML format.

The concept of availability monitoring of monitored components can be described as an agent concept. This means that the GRMG application can run separately from the components and applications that it is monitoring. This detour means that if errors occur, you can differentiate between cases in which the components monitored in the

scenario are not available (component errors) and those in which the scenario itself is not working correctly (for example, due to communication errors or an agent that is not running) (scenario errors).

The following different scenarios exist for setting up GRMG monitoring:

- Technical Customizing for monitoring a GRMG application

You have a complete Java application with a built-in GRMG application (from SAP or programmed yourself) and want to activate the availability monitoring for Java/HTTP-compatible components or Java applications.



Note: This process is suitable for consultants and customers who want to activate GRMG monitoring for an application that is already instrumented for monitoring with the GRMG.

- Instrument the application for GRMG monitoring

You have a Java component or applications for which you want to create GRMG monitoring. You need to store all of the information (host name, application, and so on) required for an automatic GRMG request in a GRMG Customizing file. Create the messages that are to be returned in the GRMG response and create a monitor definition in the CCMS Alert Monitor.



Note: This process is primarily suitable for application developers working for customers or partners who want to equip their own components for GRMG monitoring.

For more information about this, see the following sections.

Availability Monitoring of SAP NetWeaver AS Java and of Java Applications

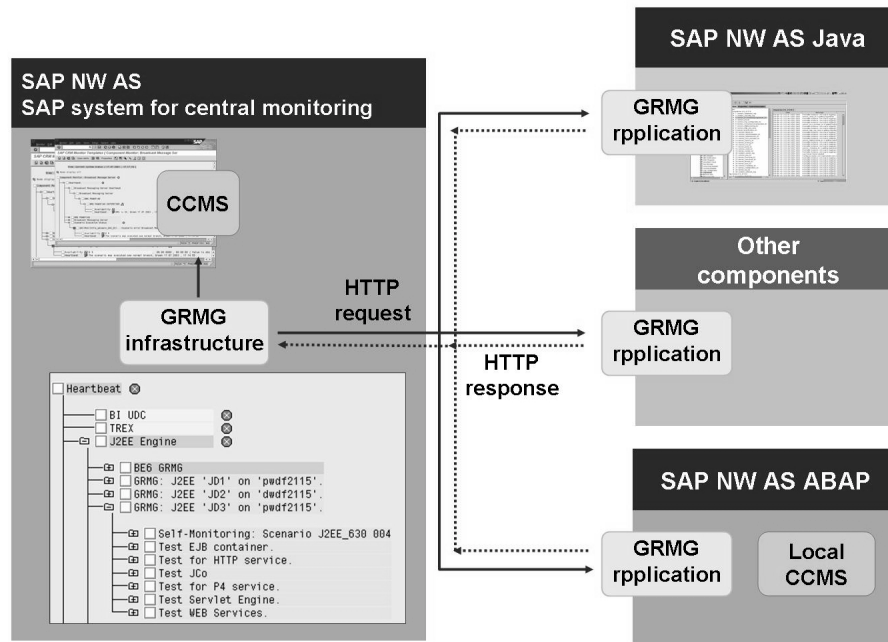


Figure 188: Availability Monitoring with the GRMG

You can use a central monitoring system to monitor the availability selected components of an SAP solution with the GRMG. The GRMG is suitable both for technical monitoring and for application monitoring. GRMG availability monitoring uses functions of the CCMS monitoring infrastructure (SAP NetWeaver AS ABAP) to store the heartbeat information. The communication is performed using HTTP POST.




Note: Heartbeat - A signal is sent by the software at regular intervals to communicate the availability (running/not running).

GRMG monitoring is performed as follows:

1. An XML message is sent from the GRMG infrastructure to the target system.
2. The GRMG application on the target system performs all of the tests for the availability monitoring of the component to be monitored or the business process step. The results of these tests are collected in the GRMG application and combined as the GRMG response.
3. The GRMG response is sent back to the GRMG infrastructure and is displayed in the Alert Monitor of the SAP NetWeaver AS ABAP as heartbeat information.

Setting Up Availability Monitoring Technically:

1. Change the *grmg-customizing.xml* file in the Visual Administrator.
 **Note:** Templates for the Customizing files are delivered with the application.
2. Upload the *grmg-customizing.xml* into the central monitoring system:
 - Automatically using an agent (SAPCCMSR)
 - Manually using transaction GRMG (central monitoring system)
3. Start the GRMG scenarios for availability monitoring

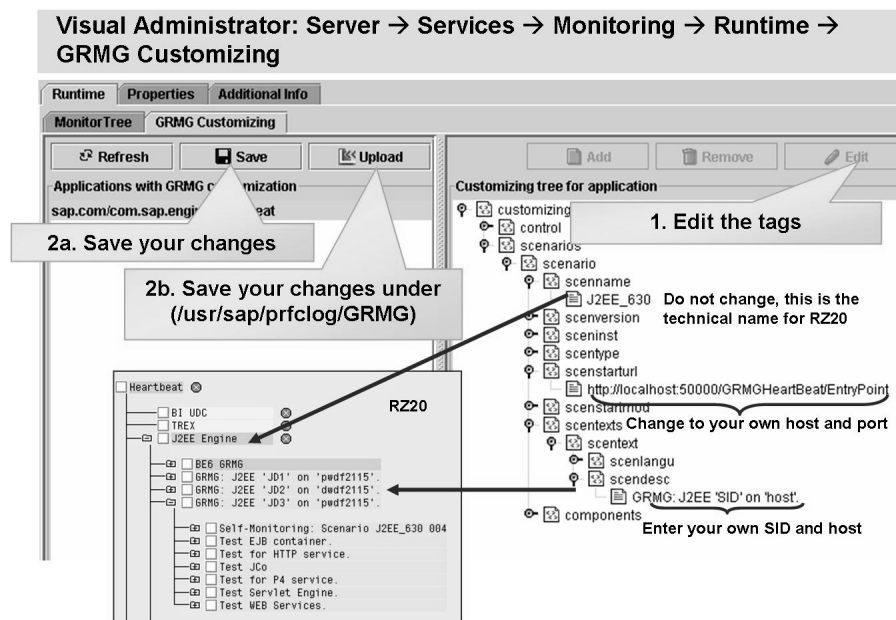
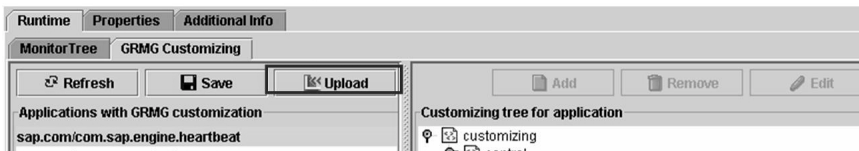


Figure 189: Editing customizing.xml in the Visual Administrator (Step 1)



Automatic Upload Using an Agent



Manual Upload

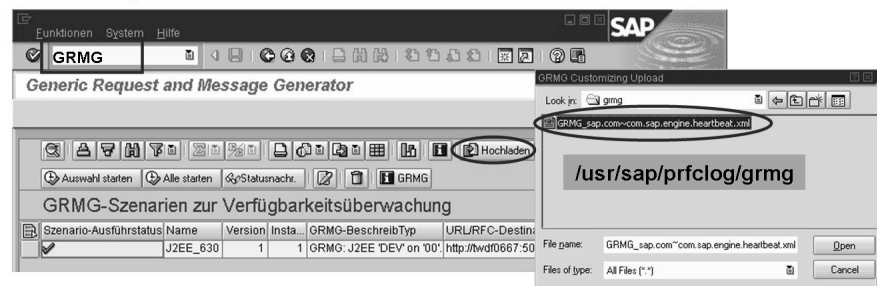


Figure 190: Upload GRMG Customizing File (Step 2)



Hint: A report that runs every hour informs the agent about the GRMG scenario. It can therefore take up to 59 minutes before the agent transfers data to the central monitoring system for the first time and the GRMG monitor is created.



Figure 191: Starting the GRMG Scenarios (Step 3)

You can use the Alert Monitor (transaction RZ20) to display availability data. In transaction RZ20, choose the *SAP J2EE Monitor Templates* monitor set. Start the *Heartbeat* monitor there.



RZ20 → SAP J2EE Monitor Templates → Heartbeat → J2EE Engine

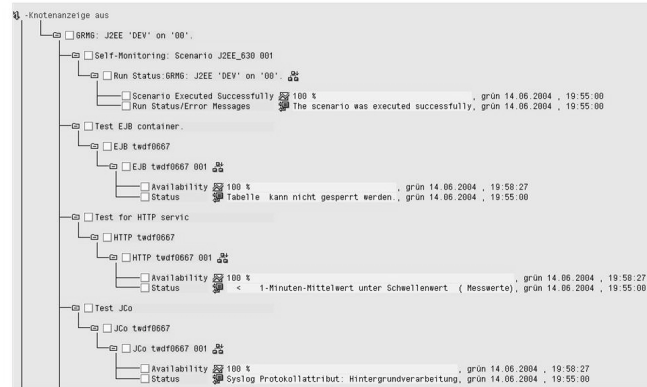


Figure 192: Availability (GRMG): Display in RZ20

If a scenario is running correctly, the components monitored by the scenario are displayed. For each monitored component, you can see the availability as a percentage, by default, averaged over the last 15 minutes, and the status with status messages that are returned by the GRMG application. To display the messages in the Alert Monitor, choose the Details button. If an error occurred in the scenario, the scenario would become red and the subtrees for the monitored components would appear colored white.

Instrumenting Availability Monitoring for Java Applications

The following process provides an overview of the steps required to instrument an application for availability monitoring with the GRMG. The following steps are a **Roadmap for Developers**:



- Design your GRMG scenario (which applications, components, processes, and so on).
- Create the messages that are to be returned in the GRMG response.
- Create a template for the GRMG Customizing file.

The GRMG Customizing file contains all information required about the scenario, the monitored components, and the parameters that are sent with the GRMG request for the components.

- Implement the GRMG application.

The GRMG application receives the GRMG request with all transferred parameters from the GRMG infrastructure, executes the availability checks, and returns the result to the GRMG infrastructure as the GRMG response.

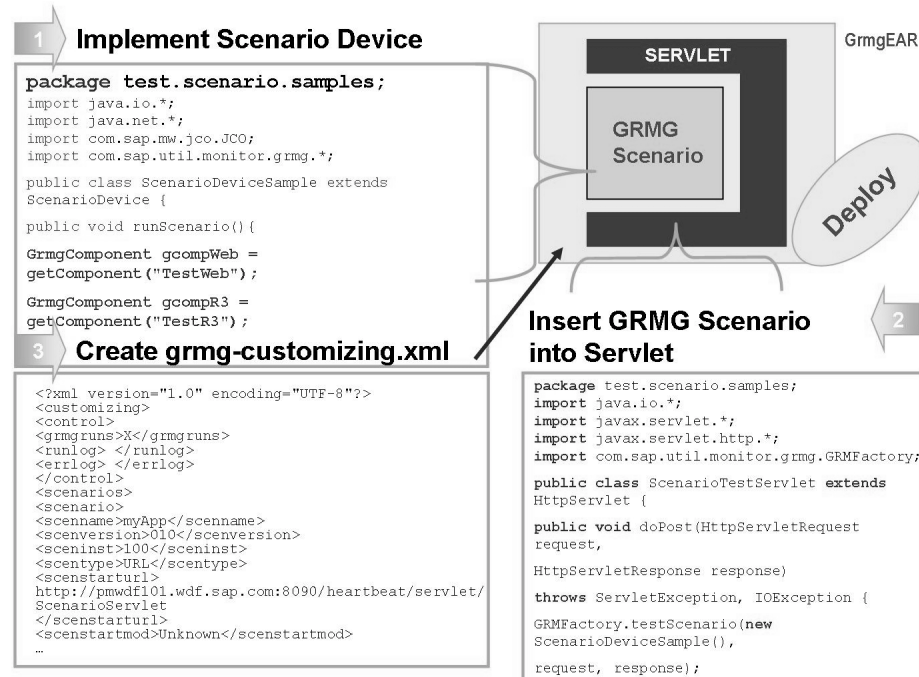


Figure 193: Creating a GRMG Application



Hint: Scenarios with different software components (especially if there are no active data suppliers available for these components) and Web-based business scenarios are typical examples of applications that you can usefully monitor with the GRMG.

Exercise 22: Availability Monitoring

Exercise Objectives

After completing this exercise, you will be able to:

- Configure availability monitoring with the GRMG

Business Example

You are using SAP NW AS Java and want to be notified as quickly as possible if a Java application or technical Java component is not running. In this case, it is useful to configure an availability check using the GRMG.

Task: Availability Monitoring

Configure an availability check using a heartbeat for the SAP NW AS Java.

1. Call the Monitoring service in the Visual Administrator, check the parameter *scenstarturl* and maintain the parameter *scendesc* of the GRMG application *sap.com/com.sap.engine.heartbeat*.
2. Start the manual upload. Then check whether the scenario that you have just loaded is visible in transaction GRMG.



Caution: Log on to the SAP system:

Start the SAP GUI for Windows on the operating system of your training host.

3. Start your scenario in transaction GRMG and then check in the Alert Monitor (transaction RZ20), whether it is delivering values.

Solution 22: Availability Monitoring

Task: Availability Monitoring

Configure an availability check using a heartbeat for the SAP NW AS Java.

1. Call the Monitoring service in the Visual Administrator, check the parameter *scenstarturl* and maintain the parameter *scendesc* of the GRMG application *sap.com/com.sap.engine.heartbeat*.
 - a) Start the Visual Administrator from the directory *G:\usr\sap\<SID>\<instance>\j2ee\admin* with the executable file “go.bat”. Log on with a user and password and the appropriate port (your instructor will provide the user, password, and port information).
 - b) Navigate to *Server* → *Services* → *Monitoring*. Proceed as described in the figure *Editing customizing.xml in the Visual Administrator (Step 1)*, and select the GRMG application *sap.com/com.sap.engine.heartbeat*. Check whether the correct server name and port for your system is entered in the *scenstarturl* parameter. You can change the parameter *scendesc* and enter a different name.
 - c) Save your entry with the *save* button and then choose the *upload* button to store the customizing file in the directory */usr/sap/prfclog/grmg*.
2. Start the manual upload. Then check whether the scenario that you have just loaded is visible in transaction GRMG.



Caution: Log on to the SAP system:

Start the SAP GUI for Windows on the operating system of your training host.

- a) Log on to the operating system of your training system and start the SAP GUI there. You can log on to the SAP system with the user **adm200-xx** (where xx is your group number), and the appropriate password.
- b) Proceed as described in the figure *Upload GRMG Customizing File (Step 2)*. To do this, log on to your SAP system and start transaction GRMG. Perform the manual upload.

Continued on next page

3. Start your scenario in transaction GRMG and then check in the Alert Monitor (transaction RZ20), whether it is delivering values.
 - a) Select your scenario in transaction GRMG, select the *Start/Stop* button, and choose *Start scenario*.
 - b) Now open transaction RZ20 and navigate to *SAP J2EE Monitor Templates* → *Heartbeat* → *J2EE Engine*. Availability information should be displayed there.



Lesson Summary

You should now be able to:

- Describe how an availability check using the GRMG works technically
- Configure an availability check
- Explain which steps a developer must perform to create a GRMG-compatible application

Related Information

- service.sap.com/monitoring
- service.sap.com/javamonitoring

Lesson: Appendix: Statistics and the Performance Trace

Lesson Overview

SAP NetWeaver AS Java provides various trace options and writes statistics. You can display the statistics in the Alert Monitor of SAP NetWeaver AS ABAP.



Lesson Objectives

After completing this lesson, you will be able to:

- List the different trace options
- List the different statistics options
- Discuss how traces are activated and where they are displayed
- Display the most important (Java) statistics in SAP NetWeaver AS ABAP

Business Example

You are using SAP NetWeaver AS Java and want to perform a performance analysis. In this case, you can activate traces or display statistics.

Performance Data

Performance data is divided into performance statistics and performance traces. Performance statistics are designed to provide a general overview of the performance data. Performance traces, on the other hand, provide more detailed information about the data flow.

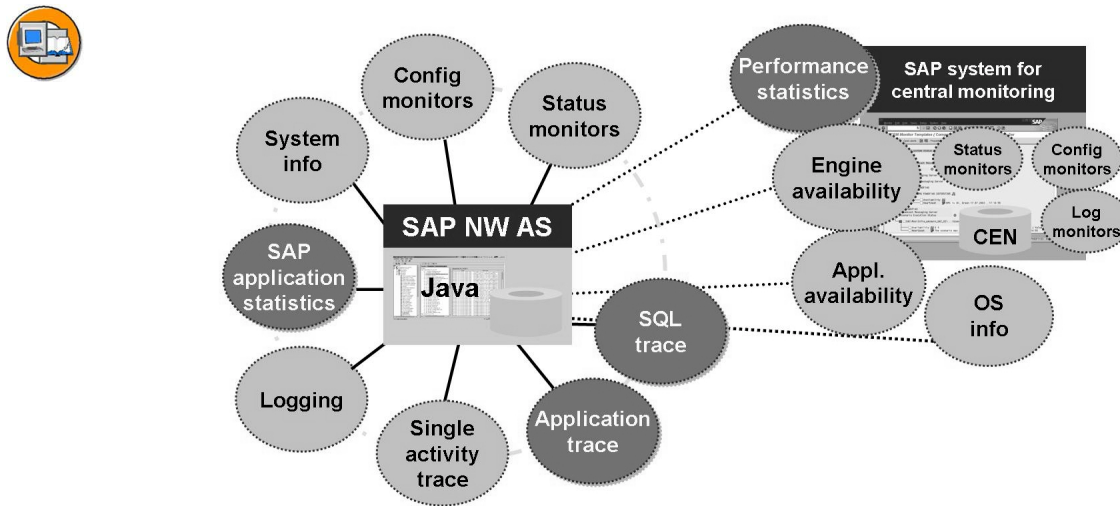


Figure 194: Overview: Traces and Statistics in SAP NetWeaver AS Java

Performance statistics include:

- Java Application Response time Measurement (JARM) for all SAP Java applications
- Distributed Statistics Records for the NetWeaver Web AS Java/EJB Container/Web Container

Thanks to Distributed Statistics Records (DSR) aggregated statistical data is provided in a central monitoring system.

The following traces are grouped together under the term performance traces:

- Single Activity Trace (SAT)
- Performance trace
- Application Trace
- SQL Trace
- Logging API trace

SAP provides many different traces and statistics in the SAP NetWeaver AS Java area, of which the administrator mainly uses the following in practice:

- DSR data in transactions ST03G and STATTRACE in SAP NetWeaver AS ABAP
- JARM data in SAP NetWeaver AS Java (JARM Viewer and Single Activity Trace)

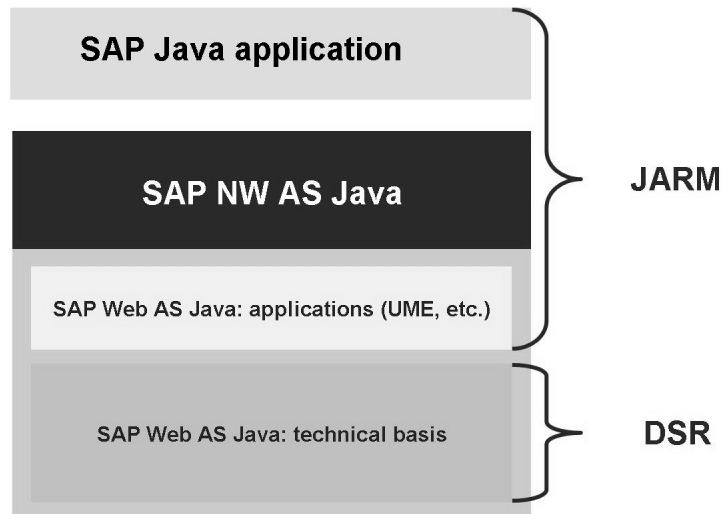


Figure 195: DSRs and JARM

You can use Distributed Statistics Records (DSRs) to obtain performance information for the technical area of SAP NetWeaver AS Java. SAP Java applications for which JARM is implemented can provide information about the response times of an application using JARM data. The following sections provide information about tools that you can use to display JARM and DSR data.

Performance and Tracing Data for SAP NetWeaver AS Java

You can view the JARM data in the Visual Administrator of SAP NetWeaver AS Java.

Java Application Response Measurement (JARM)

Java Application Response Measurement (JARM) is a method of collecting the response times of a Java application. The developer determines whether JARM data is collected for an application. The data provides an overview of components that are controlled by a thread, and at which time was spent for the component. JARM provides not only the response time, but also the user that created the request and the quantity of data transferred. The slowest response times for requests are also made available, for which you can use response time or data transferred as sort criteria. This function is intended for performance measurements and for discovering problems in the production environment. The JARM data is displayed in the Visual Administrator in the *Performance Tracing* service.



Note: In the *Performance Tracing* service, under *runtime*, you need to refresh on the *JARM* tab page to see the most recently collected data.

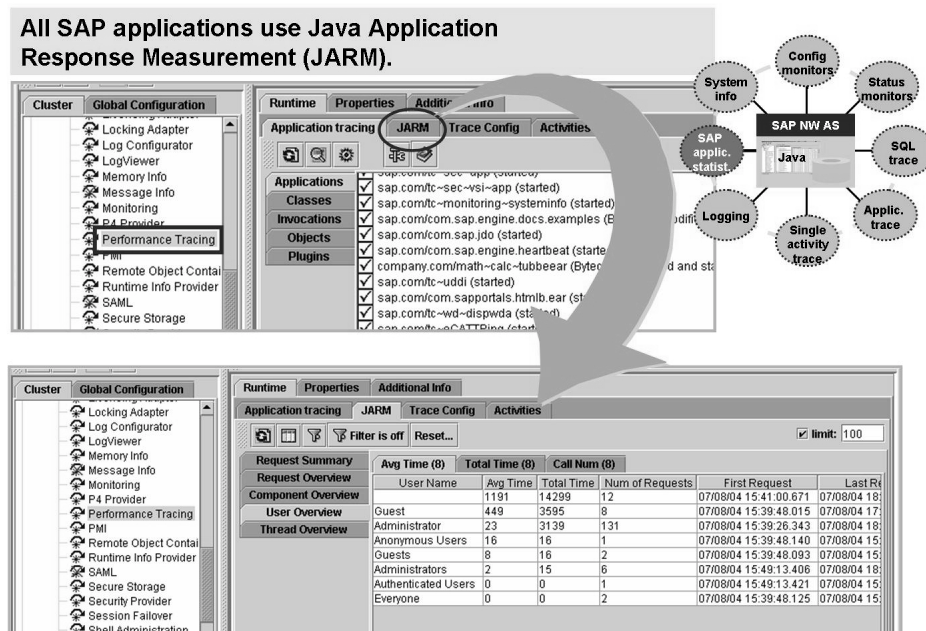


Figure 196: Java Application Response Measurement (JARM)

JARM monitoring can be activated and deactivated in the Visual Administrator. After logging on, navigate to *Server* → *Services* → *Performance Tracing* → *Properties*. Select the property *jarm/switch* and change the value to “on” or “off”.



Note: By default, JARM monitoring is active.

Some JARM data is transferred to the central monitoring system. This is the data displayed under *Monitoring Service* → *Root* → *Performance* → *Application Responsetime*. After you have installed the agent, you can view the data in the Alert Monitor (transaction RZ20) under *SAP J2EE Monitor Templates* → *Engines* → *<SID>* → *Performance*.

Single Activity Trace (SAT)

The **Single Activity Trace (SAT)** is used to trace individual (user) activities, which are running distributed across multiple components. If a performance problem occurs, use SAT to start more detailed analysis in a component. The Single Activity Trace is based on data that is provided by **Java Application Response Measurement (JARM)**. This means in practice that all SAP Java applications that are instrumented with JARM can write an SAT.

A separate Single Activity Trace is written on each component. The traces are combined using passports. Each request receives a passport, which is transferred to all of the components involved.

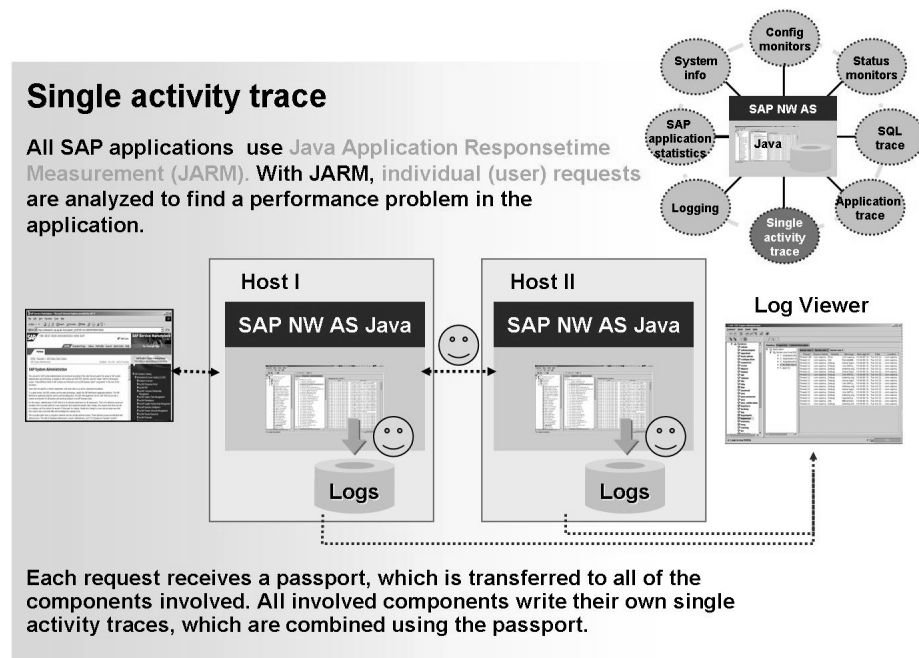


Figure 197: Single Activity Trace 1/2

All user actions that are processed within a called application are recorded. If, for example, you create users in the User Management Engine (UME), performance data is recorded for the logon process for the UME, for the call of the “Create User” application, and for the saving of the details.



You can activate the SAT trigger for the single activity trace for all or specific users. The trace file is displayed in the Log Viewer.

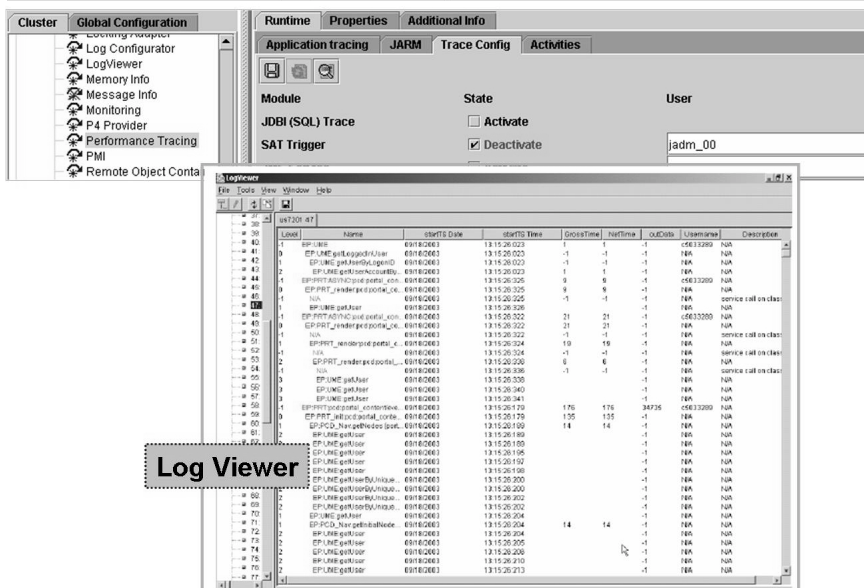


Figure 198: Single Activity Trace 2/2

The SAT data is automatically written to a trace file for every request and component using the SAP Logging API. You can display this trace file using the Log Viewer.



Note: There is a separate log type for SAT, the *SAT Trace Format*.

Performance Data in the Central Monitoring System

SAP NetWeaver AS Java collects statistics and trace data itself. This data is based on Distributed Statistics Records (DSRs) and can be transferred to an SAP ABAP system using an agent.

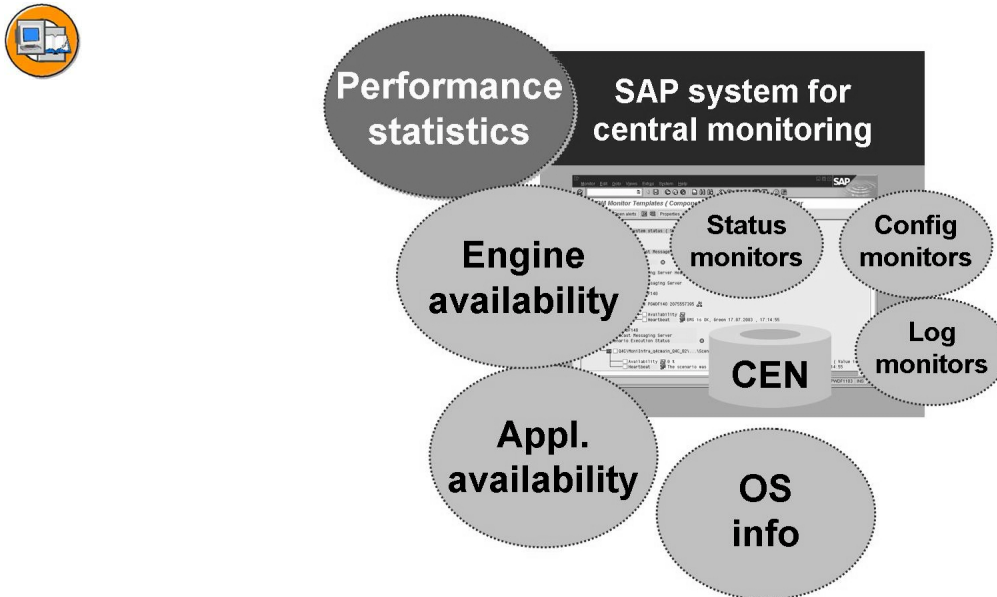


Figure 199: Performance Data in the Central Monitoring System

You can display the following performance data for SAP NetWeaver AS Java:

- **Global Workload Monitor Transaction ST03G**
Displays aggregated data that you can use to evaluate performance.
- **Performance Trace: Transaction STATTRACE**
Displays performance data in raw format (higher granularity). You can use this data to analyze problems.

The difference between the performance trace and the Global Workload Monitor is the type of data displayed and the way that the data is displayed. While the Global Workload Monitor displays aggregated data, the performance trace provides a more detailed view, and can therefore, for example, trace actions that belong to a single business process across system boundaries. DSRs are always written, while you need to explicitly activate the performance trace.

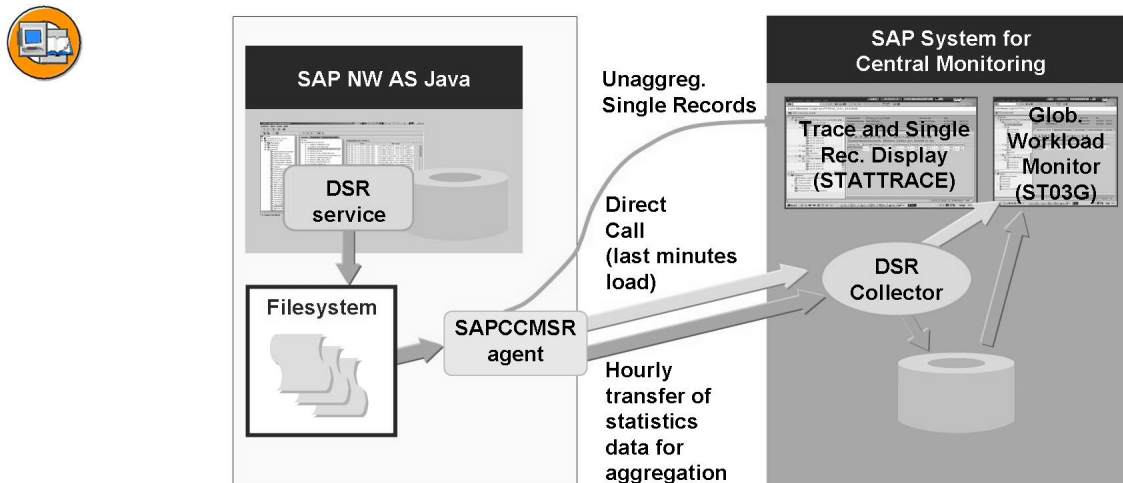


Figure 200: Generation and Transfer of Performance Data

Statistics and trace information is generated in the Visual Administrator by the *Distributed statistics* service (DSR service), and is sent to an SAP NetWeaver AS ABAP using agent. The data can either be transferred to the DSR (Distributed Statistics Records) for aggregation or can be sent directly to the Performance Trace. With the route through the collector, the statistics records collected by the hourly collector run are stored in the database and displayed in the Global Workload Monitor (transaction ST03G). The raw collected data is available to you in transaction STATTRACE.

Global Workload Monitor (ST03G)

Statistics records are generated to monitor the performance of an SAP system. They provide information about workload and about the resources that actions use in the system. Previously, it was only possible to create statistics records for an SAP NetWeaver AS ABAP and to monitor these with transaction ST03N. This concept was extended with Distributed Statistics Records (DSRs). You can use the DSRs to analyze statistical data for SAP NetWeaver AS ABAP, and also for systems that are not based on SAP NetWeaver AS ABAP. The statistics records are provided for the entire landscape, even across component boundaries. You can display the DSRs in the Global Workload Monitor (transaction ST03G).

When communicating with other components, components that write statistics records also send data for the statistics record (the passport), meaning that it is possible, for example, to trace the initiator of an action or the data flow of a business process, even across component boundaries.



Note: The passport allows the creation of Distributed Statistics Records (DSRs) and performance traces across component boundaries.

The DSRs are first saved locally on the relevant component and transferred hourly to a monitoring system using CCMS agents. In this monitoring system, the aggregated statistics data is stored in a performance database and regularly reorganized.

The Global Workload Monitor displays the statistics data aggregated by the collector. You can use the functional trace (transaction STATTRACE), on the other hand, to display unaggregated raw statistics data (single records) for SAP ABAP systems and non-SAP-ABAP systems from complex system landscapes. The functional trace provides a finer granularity. With the functional trace, you can trace actions that belong to a single business process across system boundaries.

You can perform the following analyses, among other things, in the Global Workload Monitor:

- What is the workload of individual actions?
- How is the workload distributed across the individual hours of the day?
- Which action steps are showing the longest response and waiting times?
- What load data is collected when external components are called?
- What is the workload of individual users, and which actions has a user performed?
- What load is generated in a component due to actions of external components?



Performance information for SAP NW AS Java and SAP NW AS ABAP is collected in Distributed Statistics Records (DSR) and made available centrally in transaction ST03G.

Workload overview

Action profile

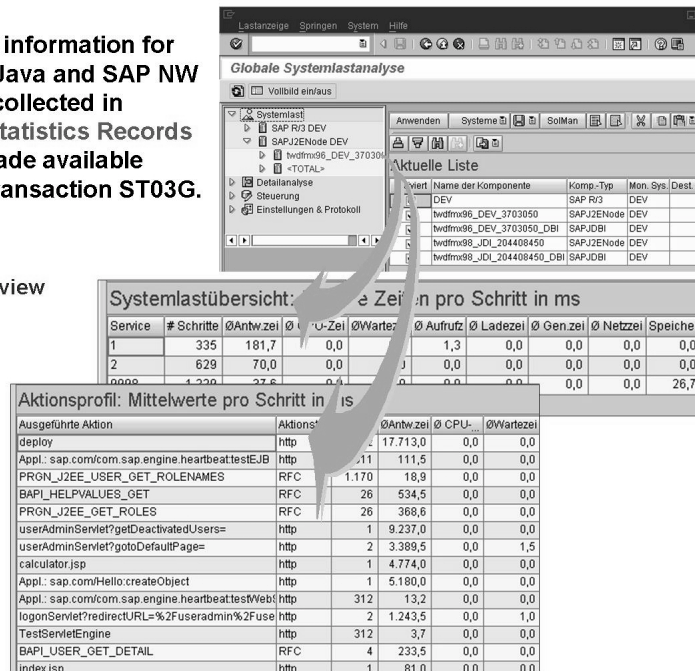


Figure 201: Distributed Statistics Records: ST03G

Statistics data for non-SAP-ABAP systems is initially collected in the component itself and is then transferred with an agent to the monitoring system with which the agent is registered. In this monitoring system, the data is further processed, aggregated, and stored in a performance database. This is done by the workload collector, which runs once an hour, by default. The collector is started by the job SAP_COLLECTOR_FOR_NONE_R3_STAT. By default, this job runs hourly and is one of the standard SAP jobs.

There are many ways in which you can influence the type and scope of the data collection and data storage. ST03G contains functions for database and parameter maintenance for the DSR collector and the DSR performance database:

- Display and delete the contents of the performance database

Statistics data for systems that are not based on SAP NetWeaver AS ABAP is, like the SAP ABAP statistics records, stored and aggregated in a performance database. You can display the data records of this DSR database that are available locally in a table.

- Control the reorganization of the performance database

With this procedure, you define how long the daily, weekly, and monthly statistics for the individual aggregates are to be retained in the performance database. These parameters apply in the same way to all components and component types.



Note: The reorganization is usually performed once a day by the job SAP_REORG_NONE_R3_STAT_DB. A delete operation is recorded in the collector log. The reorganization job is started by the job SAP_COLLECTOR_FOR_NONE_R3_STAT. By default, this job runs hourly and is one of the standard SAP jobs.

- Display collector logs

You can display the messages of the collector for the non-SAP(-ABAP) statistics and of the various component collectors, and filter them by different types (information, warning, error, or fatal).

- Set parameters to control the DSR collector

You can define general parameters to control the collector, depending on the component type or agent, here. You can define the retention period for the statistics files (by default, 48 hours). This is the (raw) data that is stored locally in the relevant component.

- Set parameters for statistics generation

You can use this function to activate or deactivate different statistics for each component type and to set parameters that affect the formatting of the statistics for response time distribution and availability.

- Display the application log

The application log contains error messages of the Global Workload Monitor itself. Among other things, RFC errors to other systems are recorded here. The structure of the page is identical to that of the page with the messages of the collectors.

- Define the systems to be monitored

Use this function to define which components are to be monitored using the Global Workload Monitor.

For more information about operating transaction ST03G, attend training course ADM315.

Performance Trace (STATTRACE)

You use the performance trace from the central monitoring system if you discover anomalies during the performance analysis. The performance trace is known as the functional trace in SAP NetWeaver AS ABAP and is displayed using transaction STATTRACE.

Activate the performance trace when you require duration information to be written. If, for example, the Global Workload Monitor (transaction ST03G) shows that the average response time of an application is too high, you should activate the trace to find the cause. The performance trace is based on DSRs, but you can activate the collection of additional performance data for each module. Compared to DSRs, the performance trace has the advantage that the SAP NetWeaver AS Java appears transparently. In practice, this means that with Distributed Statistics Records (DSRs), only the duration from entering to leaving the SAP NetWeaver AS is measured. The performance trace itself can collect the durations for the individual modules of SAP NetWeaver AS Java (JDBI (SQL) trace, HTTP service, JMS service, EJB Container, Web Container, RMI Connector, RFC Connector). In this way, the trace provides a finer view of the processes in the SAP NetWeaver AS Java, since the area between the entry and exit points is displayed in more detail.

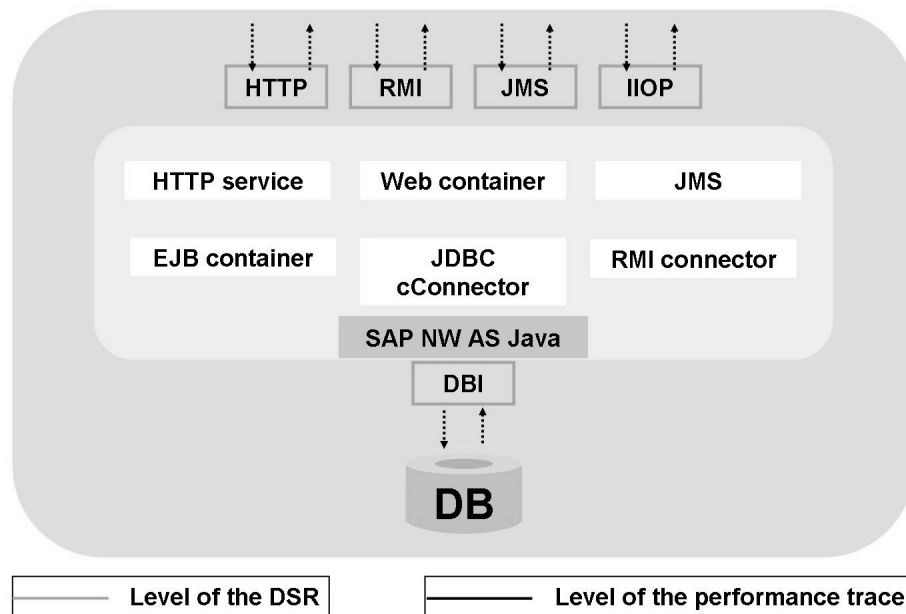


Figure 202: Performance Trace versus DSR

The trace records the durations within the modules, that is, the names of the distinct called methods and their duration are recorded for each module. The methods that are displayed is predefined. These are method calls that are characteristic for the corresponding module. The method calls are displayed with the class, method name, duration in microseconds, and whether the method was ended normally or with an exception.



Note: In the Global Workload Monitor (transaction ST03G), for example, the service type is always specified as the type of the last request. This means that if the request ran through multiple containers and finally to the EJB Container, the request is listed as an EJB request in the Global Workload Monitor. If the administrator wants to know exactly where the individual times were spent, he or she needs to activate the performance trace.

The performance trace data is transferred as raw data to transaction STATTRACE using the agent. The following prerequisites apply for the performance trace data to be written and displayed in the SAP NetWeaver AS ABAP:

- The SAPCCMSR agent must be installed
- Release of the central monitoring system: SAP NetWeaver AS 7.0
- The DSR service must be active

You can activate the performance trace using the Visual Administrator and, among other things, configure the modules of the SAP NetWeaver AS Java for which performance traces are to be written.



Visual Administrator:

Server -> Services -> Performance Trace -> Runtime -> Trace Configuration

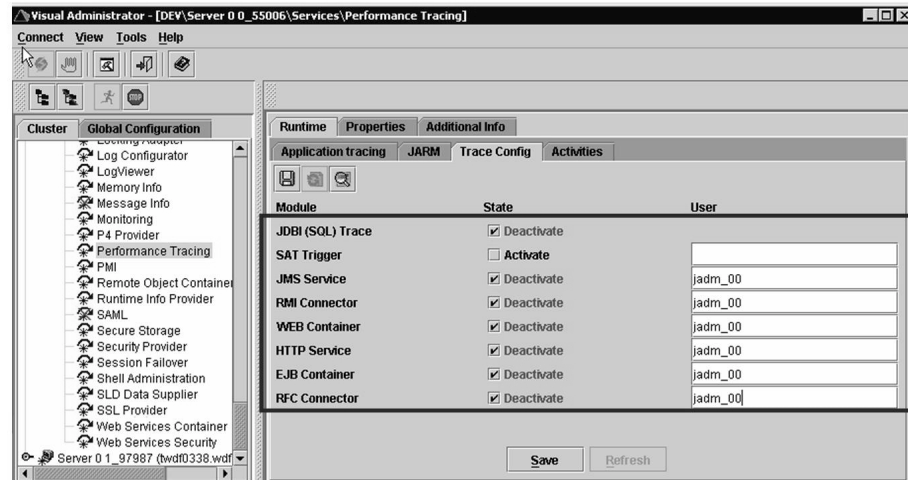


Figure 203: Activating the Performance Trace

- Service: *Performance Tracing* → *Runtime Control* → *Trace Configuration*
- Activate the relevant performance traces at the touch of a button (possibly set user filter)

You can display the performance traces in the central monitoring system using the functional trace (transaction STATTRACE).

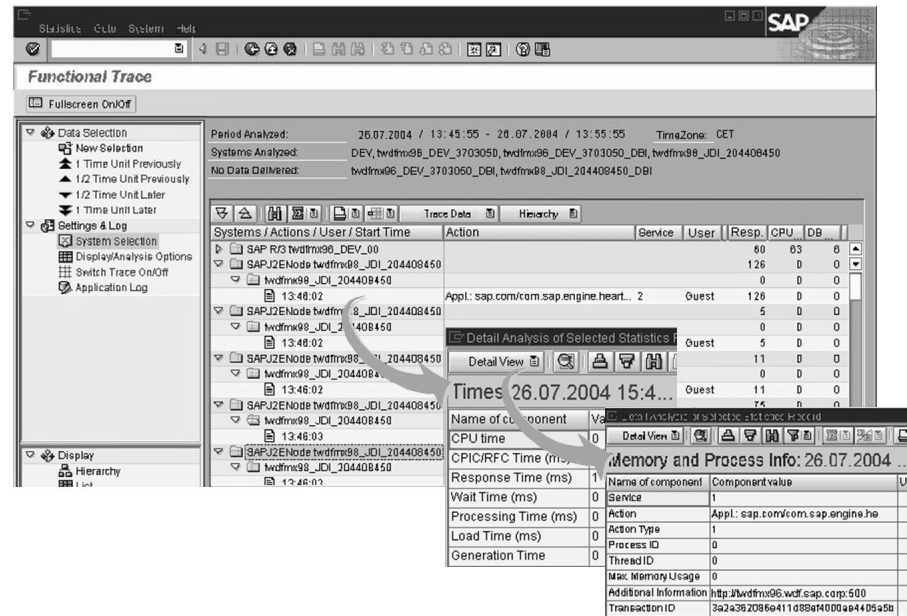


Figure 204: Transaction STATTRACE

Transaction STATTRACE offers the following functions:

- **System selection:** You can select the systems for which you want to analyze the statistics record and, if appropriate, traces.
- **Data selection:** In the data selection, you can define a period for which the statistics records are to be read. The trace is read for this period for the components specified in the system selection.
- You have various options for displaying and analyzing the trace data. You can, for example, display the statistics records sorted chronologically in a call hierarchy or as a list.

Appendix: Other Trace Capabilities

If there are performance problems, you can use the traces listed here to obtain additional, more detailed information.

SQL Trace

The SQL trace provides additional functions for certain selectable SQL statements. The SQL statements use calls to a database through open and native JDBC methods. You can activate and deactivate the SQL trace dynamically. In addition to the SQL statement text, it provides information about the time, duration, result, and input parameters of the executed statement.

An SQL trace is used in the following cases:

- For development
Developers of Enterprise Java Beans, servlets, and Java Server Pages receive information about database access that is generated using their own Java coding.
- For performance analysis
Performance problems are often caused due to inefficient database accesses. In this case, an SQL trace is used to display the SQL statements used and their duration.



The SQL Trace is activated in the **Log Configurator** service of the Visual Administrators or using <http://<host>:<port>/SQLTrace>.

The screenshot shows the SAP Log Configurator interface with the 'Enable SQL trace' button highlighted. Below it is the 'SQLTrace Evaluation' web page, which displays a table of SQL statements and their execution details.

Time	Duration (μ)	Java method Id	No.	Result	Statement
12.5405.312	6413	Direct(Connection.asAutoCommit(false))			setAutoCommit(false);(Auto.)
12.5405.642	2263	Direct(PreparedStatement.executeQuery())			SELECT DISTINCT "PR" FROM "TIME_STRINGS" WHERE "PR" LIKE ? ESCAPE '~' AND "NAME..."
12.5405.723	1069	Direct(DatabaseSetLast())	1	True	next()
12.5405.823	28	Direct(DatabaseSetLast())	1	False	next()
12.5406.023	25	Direct(DatabaseSetLast())			close()
12.5406.393	1043	Direct(Connection.asRollback())			rollback()
12.5406.634	30	Direct(Connection.asAutoCommit(false))			setAutoCommit(false);(Auto.)
12.5406.734	13	Direct(Connection.asRollback())			rollback()
12.5406.814	24	Direct(Connection.asTransactionIsolation())			setTransactionIsolation(1);

You can display the SQL trace using a special Web page.

<http://<host>:<port>/SQLTrace>

Figure 205: SQL Trace

Open SQL monitors and SQL monitors are available as a separate Web-based application. This application is deployed during the installation of SAP NetWeaver AS Java.

You can call this application using the following URL <http://<server>:<port>/SQLTrace>, such as <http://P12345:50000/SQLTrace>.

You can activate the SQL trace both in the Visual Administrator and by URL.

- Visual Administrator
Server → Services → Log Configurator → enable/disable SQL Trace
- URL
http://<host>:<port>/SQLTrace → Switch SQL Trace on/off

Application Tracing

The **application trace** is intended for **developers**, so that they can obtain additional, more detailed information about their application. The application trace is usually used during the development process. The application trace allows you to perform a quick trace without setting the Virtual Machine (VM) into debug mode, restarting the container, or redeploying the application (to deploy the application is to make it available). The application trace is a powerful tool for “on-the-fly” debugging of J2EE applications. It can perform measurements for individual Java programs.

If you activate the application trace for an application, the application is assigned “markers”, which measure the time taken for each called Java method. The application is started in the *bytecode-modified Mode* and a trace is therefore activated. As soon as the debugging process is complete, switch back to *normal mode*. The “markers” are automatically removed again by the garbage collection. Therefore, you do not have to redeploy the application.



Hint: You usually use the application trace on the development/test system and not directly in the production environment, since the application is restarted in *bytecode-modified mode* when you activate the application trace.

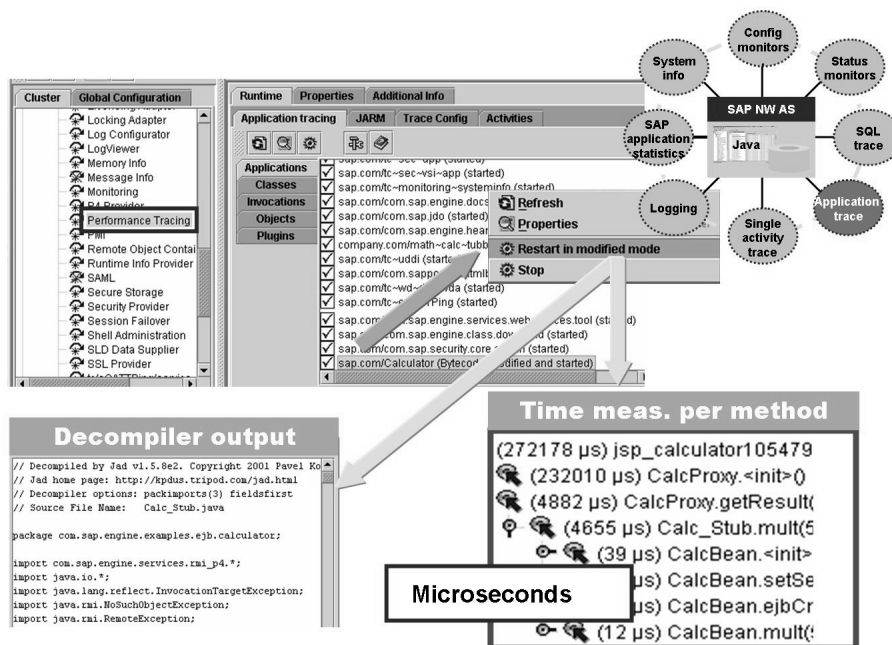


Figure 206: Application Tracing

In the Visual Administrator, you can use the *Performance Tracing* service, on the *Application Tracing* tab page, to select an application and start it in *bytecode-modified mode*. The results are displayed in the Visual Administrator. As soon as the application is used, you see all method invocations of the application that are integrated into the process flow. The time required for each method is measured in microseconds. An integrated decompiler displays the source code.

Exercise 23: Statistics and the Performance Trace

Exercise Objectives

After completing this exercise, you will be able to:

- Display the most important (Java) statistics in SAP NW AS ABAP

Business Example

You are using SAP NW AS Java and want to perform a performance analysis. In this case, you can activate traces or display statistics.

Task 1: SQL Trace

Activate the SQL Trace and search for statements which have been running for a “long” time.

1. Activate your instance's SQL Trace via the browser.
2. Open a new browser window and call the NWA in the same instance. Display your Expert view under Logs and Traces.
3. Deactivate the trace again. Deactivate the trace and search for the statements that have lasted the longest.

Task 2: Application Trace

Start the application trace

1. Start the application trace for the *SQLTrace* application from the previous task.
2. Display the measured values in the application trace.

Task 3: Optional: JARM

Performance data in the JARM Viewer

1. Display the requests with the longest response times in the JARM Viewer.
2. Display the requests with the longest network times in the JARM Viewer.

Solution 23: Statistics and the Performance Trace

Task 1: SQL Trace

Activate the SQL Trace and search for statements which have been running for a “long” time.

1. Activate your instance's SQL Trace via the browser.
 - a) Start the URL <http://<hostname>:<port>/SQLTrace> e.g. <http://twdf9999:50000/SQLTrace>, in the browser to go directly to your SAP NW AS Java's SQL Trace.
 - b) Choose *Switch on and off SQL Traces* to go to the SQLTrace Status view.
 - c) Select your instance's cluster node and choose the button *Sel. nodes: ON*. In the Status view, the node is now identified as “On”.
2. Open a new browser window and call the NWA in the same instance. Display your Expert view under Logs and Traces.
 - a) In the browser, start the URL <http://<hostname>:<port>/nwa>. e.g. <http://twdf9999:50000/nwa>.
 - b) Choose *System Management* → *Monitoring* → *Logs and Traces* → *Custom View* to go to your Expert view and select, for example, my Expert.
3. Deactivate the trace again. Deactivate the trace and search for the statements that have lasted the longest.
 - a) Select the node with the active trace and choose the button *Sel. nodes Off*.
 - b) Now choose the button *Trace Evaluation*. Your trace is now selected. Now choose the *Evaluate* button to enter the Filter view. You can set a restriction, for example to 300000 microseconds, under *Min. Duration ...* and choose *Evaluate* to display the SQL Trace. If no entries are present, set a correspondingly smaller restriction or omit it entirely.

Continued on next page

Task 2: Application Trace

Start the application trace

1. Start the application trace for the *SQLTrace* application from the previous task.
 - a) Log on the operating system and start the Visual Administrator. Your instructor will give you the appropriate user and password information. Navigate to the *Performance Tracing* service and, under *runtime*, choose the *Application Tracing* tab page.
 - b) Select the SQL trace and activate the *restart in modified Mode* function by right-clicking.
 - c) Then perform a brief *SQLTrace*.
2. Display the measured values in the application trace.
 - a) Go to the *Application Tracing* service, on the *Classes* tab page and *Invocation*. The data is displayed there.

Task 3: Optional: JARM

Performance data in the JARM Viewer

1. Display the requests with the longest response times in the JARM Viewer.
 - a) Log on the operating system and start the Visual Administrator. Your instructor will give you the appropriate user and password information.
 - b) Navigate to *Server* → *Services* → *Performance Tracing* → *runtime* → *JARM*.
 - c) Choose the *Request Overview* → *Response Time* tab page. To display the data, choose the *refresh* icon.
2. Display the requests with the longest network times in the JARM Viewer.
 - a) Navigate to *Server* → *Services* → *Performance Tracing* → *runtime* → *JARM*.
 - b) Choose the *Component Overview* → *Avg Net Time* tab page. To display the data, choose the *refresh* icon.



Lesson Summary

You should now be able to:

- List the different trace options
- List the different statistics options
- Discuss how traces are activated and where they are displayed
- Display the most important (Java) statistics in SAP NetWeaver AS ABAP

Related Information

- service.sap.com/monitoring
- service.sap.com/javamonitring

Lesson: Appendix: Solution Manager Diagnostics (SMD)

Lesson Overview

The SAP Solution Manager Diagnostics expands the SAP NetWeaver AS Java with an important and useful analysis tool.



Lesson Objectives

After completing this lesson, you will be able to:

- List the functions of the SAP Solution Manager Diagnostics
- Understand the architecture of the SAP Solution Manager Diagnostics

Business Example

You are deploying an SAP NetWeaver-based system and using Java functions in particular. For the Java functions, the SAP Solution Manager Diagnostics provides various useful tools for system analysis

Overview: SAP Solution Manager Diagnostics (SMD)

The basic prerequisite for an efficient and secure support of IT solutions is the ability to carry out a quick and efficient end-to-end (E2E) root cause analysis. The SAP Solution Manager Diagnostics (SMD) offers all these functions to analyze and monitor an SAP NetWeaver system landscape.

SMD is a part of the SAP Solution Manager 4.0. SAP Solution Manager 4.0 is based on an SAP NetWeaver AS ABAP+Java.

The SMD brings the following features:



- Central log file viewer
- Central display of configuration data
- Monitoring functions: Database, operating system
- E2E trace analysis
- Java thread dump analysis
- Java performance analysis with Wily Introscope
- Load tests with Mercury Load Runner

Infrastructure of the SMD

The SMD is a tool that reduces the time for an E2E root cause analysis.

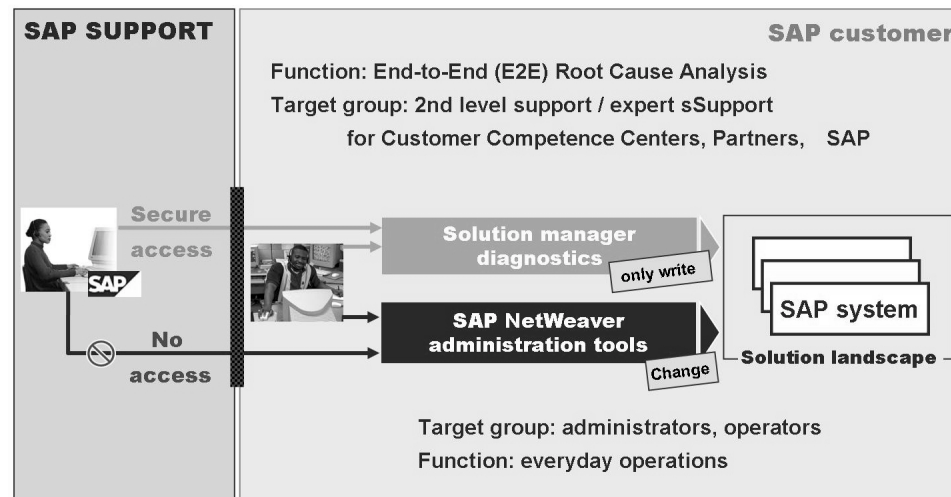


Figure 207: Overview: SAP Solution Manager Diagnostics (SMD)

The SMD is called using a browser: <http://<hostname>:<port>/smd>, e.g. <http://P12345:50000/smd>

You arrive at a central console from which all the functions of the SMD can be reached. The operating system cannot be accessed from the SMD and it is not necessary as the access is done with http and all important support and administration functions are available in the browser interface. The SMD cannot use SAP NetWeaver Administration tools that have changing access on the landscape systems.

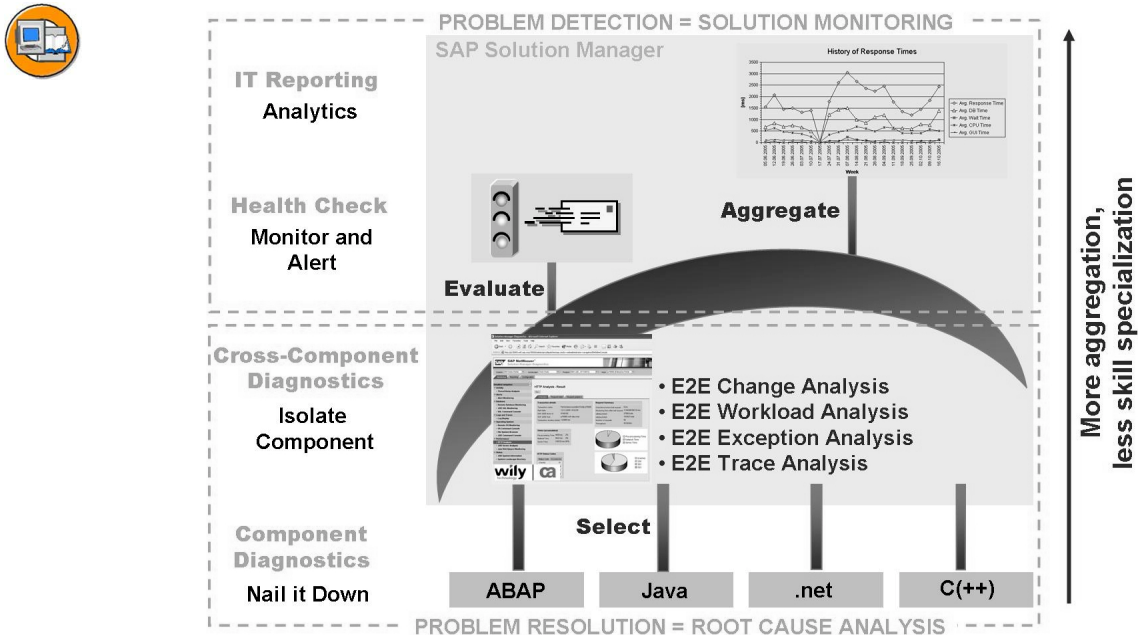


Figure 208: Solution Manager

The SAP NetWeaver Administration tools (SAP NetWeaver Administrator, Config Tool, ...) are generally used by the administrators and operators for customers for daily operation (starting and stopping, configuration changes, ...). The Solution Manager or SMD has a different focus and target audience. The target audience is defined as support (first or second-level support) for partners, customer competence centers or SAP. The SMD offers you the option of performing an E2E root cause analysis and allows tracing, performance optimization and the tracking of configuration changes.

Nowadays, system landscapes consist of different components across which elements of business processes are distributed. When analyzing problems, it is no longer sufficient to consider just one component. In such complex landscapes, it is necessary to identify the component which is causing the problem. The SAP Solution Manager adopts this approach in SAP landscapes. These “cross-component diagnostics” are summed up by the term “end-to-end” which is intended to make it clear that the analysis extends from the start of the business process (e.g. entry of the user's data) through to its end. Once the component responsible for the problem has been identified, the problem can be further analyzed in this component in the form of “component diagnostics”. SAP Solution Manager 4.0 provides tools for the “Cross-Component Diagnostic”, the “Health Check” and the “IT Reporting” but not for the “Component Diagnostic”. These tools are often referred to as “Solution Manager Diagnostics” (SMD).

Architecture of the SMD

The SMD runs in a SAP Solution Manager 4.0 that is installed as an add-in system (ABAP + Java).

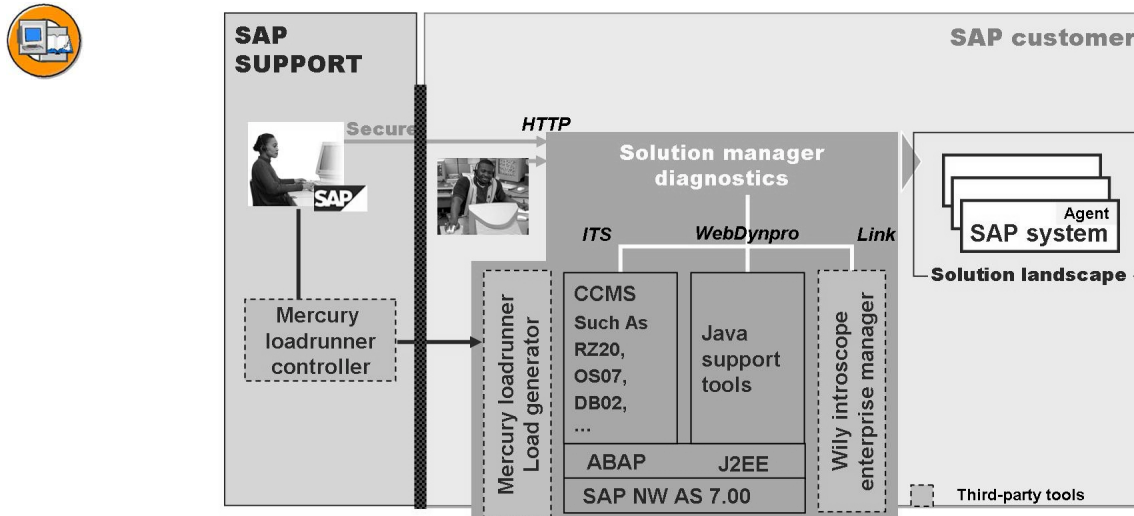


Figure 209: Architecture of the SMD

The SMD uses a number of different components:

- SAP NetWeaver AS ABAP
- SAP NetWeaver AS Java (Component Analyzer, ...)
- Wily Introscope
- Mercury Load Generator

The SMD accesses the **SAP NetWeaver AS ABAP** via the integrated Internet Transaction Server (ITS) and allows you to display monitoring data. Remote systems are connected using the CCMS agent technology.

Java support tools such as the *Component Analyzer* are available in the **SAP NetWeaver AS Java** area of the SMD.



Hint: Starting with SAP NetWeaver 04 Support Package Stack 10, the **Component Analyzer** will be automatically installed on your SAP NetWeaver AS Java system. The Component Analyzer collects configuration and file information that is requested by the SMD. From a technical point of view, it reads all data from the file system (such as ini files) and stores the information in XML files. The Solution Manager Diagnostics triggers the upload of these files and the CCMS agent transfers the files via RFC.

The **Wily Introscope** tool is integrated in the SMD and is used for performance analysis and optimization for the SAP NetWeaver AS Java and applications running on it. Wily Introscope is a client/server application. The server share is called *Introscope Enterprise Server*. The client share is called *Introscope Agent*.

The **Mercury Load Generator** tool belongs to the SMD elements as well. This tool is used to generate load for web-based applications to identify performance bottlenecks such as for a GoingLive check.



Hint: SAP has signed an agreement regarding the distribution of the two third-party tools.

The systems to be monitored must meet certain requirements:

- SAP NetWeaver AS Java 6.40 SP Stack 10
- SAP Enterprise Portal EP6 SP2 (Portal Patch 5 + Build 2, SAP NetWeaver AS Java SP25, Startup Framework 6.40)

Functions of the SMD

In the SMD application, you can select a defined solution, a landscape and a product in the *Landscape Navigation*. Depending on the selected product, the *Detailed Navigation* will change and display only the E2E root cause analysis tools that are required for the product.

SMD provides *System Analysis Tools* for all SAP solutions. Analysis tools for the database, operating system and various consoles (secure operating system command console, Java command console, database command console, ...) are available. A file system browser and the LogViewer exist as well.

The *Java Analysis Tool* is available for all Java-based applications (SAP EP, XI, ...) that run on the SAP NetWeaver AS Java. Wily Tech Introscope (performance optimization) belongs to these tools and can be used for SAP EP, SAP XI, SAP NetWeaver AS Java and Web Dynpro application, for example. Additionally, Java thread dumps can be analyzed and the Logviewer and the performance trace are available in the SMD.

The SMD also offers functions in the *Web Analysis* area. You can use the E2E trace analysis.

The Mercury Loadrunner can run load tests.

There are also component-specific analysis tools for SAP EP and SAP XI.

The SMD is in principal divided into the three large areas *Monitoring*, *Reporting* and *Configuration*. The *Monitoring* area offers many different monitoring tools. You can perform configuration steps for the SMD in the *Configuration* area. The *Reporting* area provides functions for *Statistics*, *Configuration* and *File System Administration* and *Software Change Management* in Detailed Navigation Reporting.

Detailed Information about Selected Functions in the SMD

The SMD is subdivided into a number of areas, i.e. *Exceptions*, *OS and DB*, *Configuration*, *Traces*, *Workload* and *Availability*.

If you choose “OS Command Console, File System Browser” in the Detailed Navigation under **OS and DB**, various tools will open with which you can run predefined operating system commands or have read access to the file system structure.

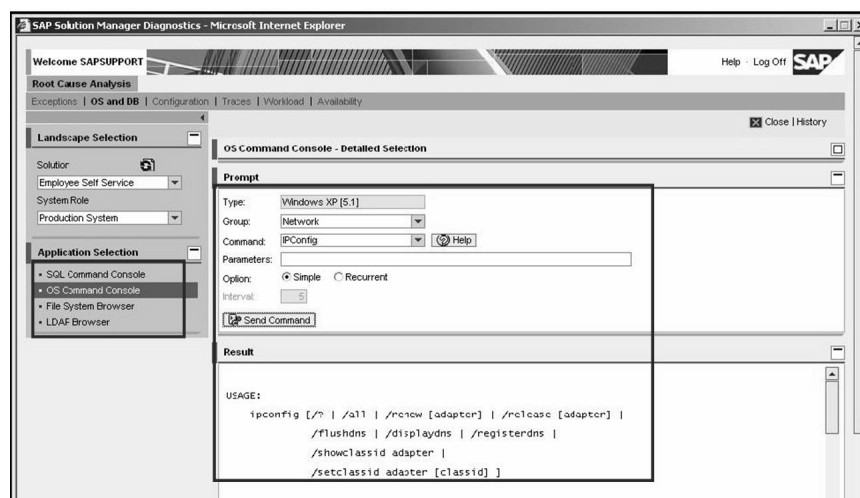


Figure 210: OS Command Console, File System Browser

Under the entry *Root Cause Analysis* → *Workload*, the Detailed Navigation displays the **E2E Workload Analysis** with, for example, the “Thread Dump Analysis”. The thread dump analysis can be used to start and analyze JVM-based thread dumps for individual Java nodes or for the entire system. The “Java Memory Analysis” displays the JVM garbage collection in graphical form.

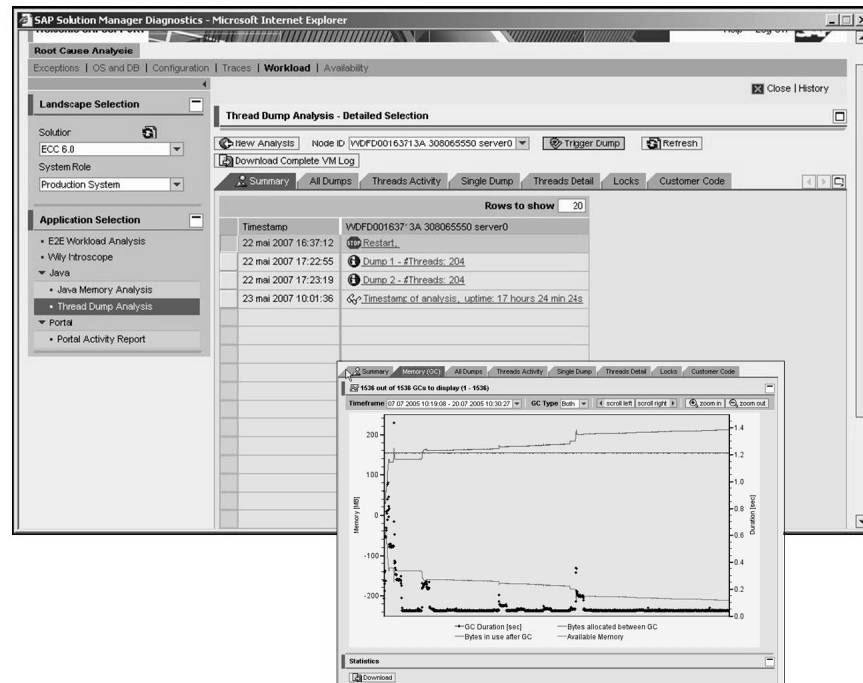


Figure 211: Thread Dump Analysis

The *Portal Activity Report* is also integrated here. There you can see statistics data for users. If you use a Portal, additional statistics data for iViews and pages is available. The *Web Content Reporting (WCR)* service waits for events that the Portal Runtime creates while processing pages and iViews. The service analyzes and collects the data. The data can also be exported for further processing. In the reporting view you can see the number of named or anonymous users. The user names are not displayed in the reporting view but are stored in the database.

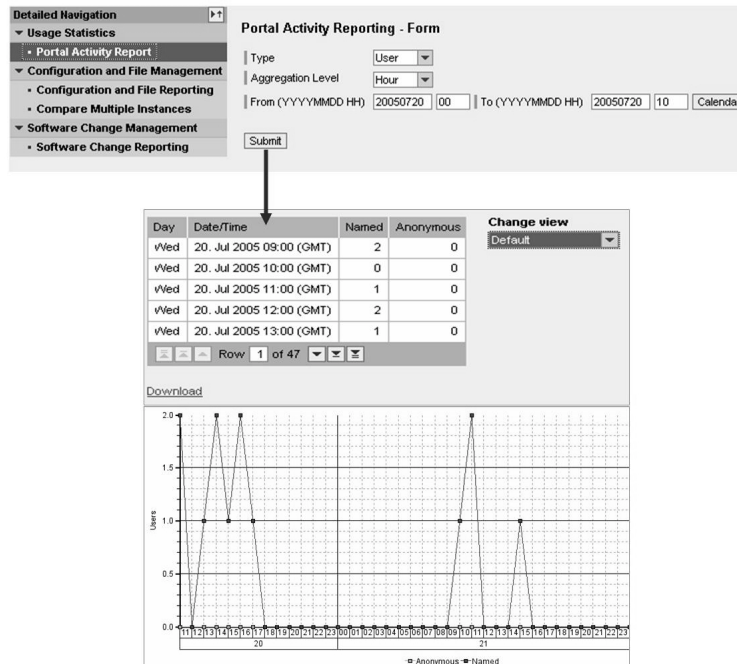


Figure 212: Portal Activity Report

Another area takes the form of the **E2E Trace Analysis** which contains the trace of the same name.

HTTP requests that are sent from a client to the server and the HTTP responses are monitored for the **E2E Trace Analysis**. Client rendering actions are entered as well. A plugin must be present in the browser (client) to measure the activities. This makes it possible to describe an XML file. HTTP logging is active on the server side and writes the trace information to a trace file. The XML file (client) and the trace files (server) are uploaded to the SMD and prepared for evaluation. This is displayed in the SMD graphically or in a table. The HTTP analysis overview shows which areas have problems. The table view provides detailed performance data for each individual request.

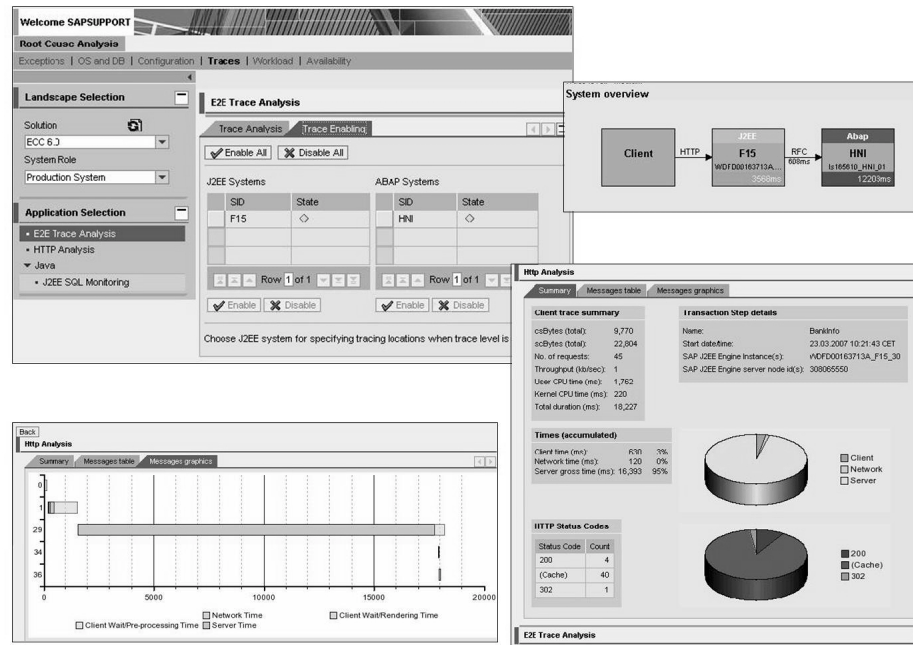


Figure 213: E2E trace analysis

In the **Exceptions** area, you see the “Logs and Traces” with which you are already familiar from the SAP NetWeaver Administrator.

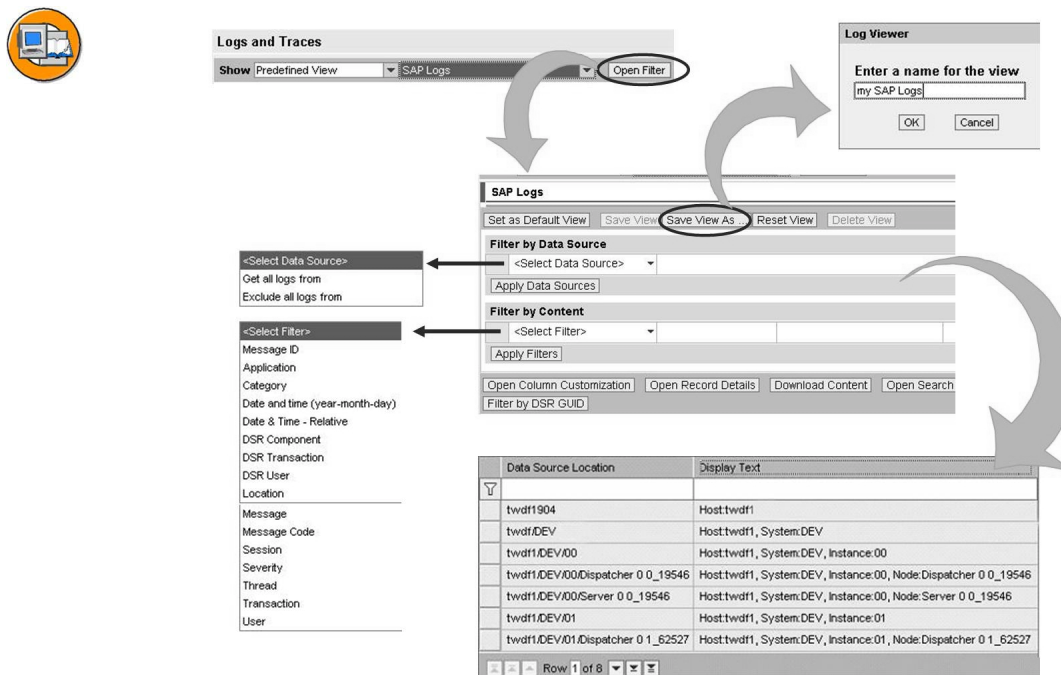


Figure 214: Logs and Traces

The **Configuration** area allows you to compare multiple instances of a cluster with each other and to use reporting functions for the configuration changes. Configuration information for all non-ABAP and ABAP components is collected and displayed in the SMD so that you can navigate to the configurations. A history of the configuration settings based on daily snapshots is available in the SMD. The active configuration setting for a certain point in time can also be displayed.

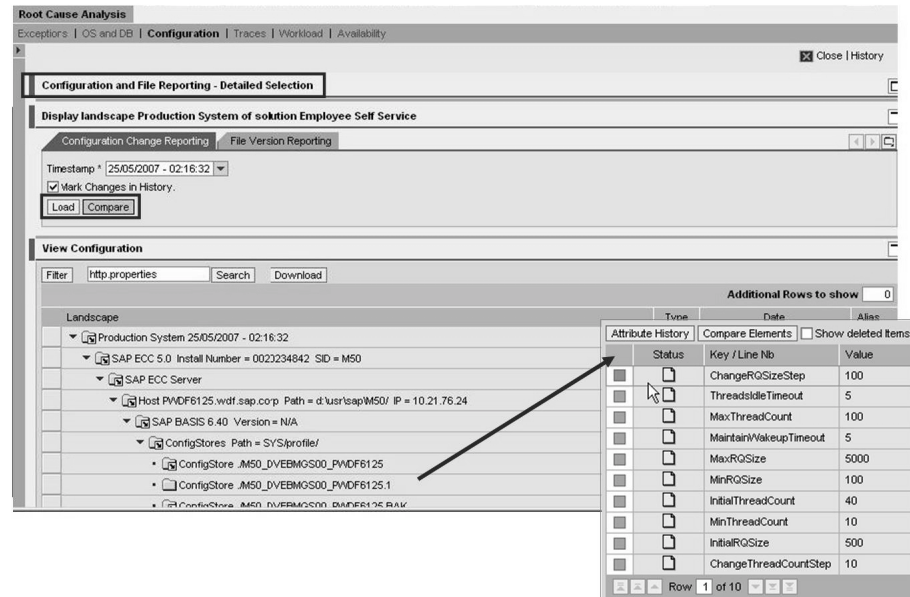


Figure 215: Displaying the Active Configuration

You can also compare components in any landscape. The SMD displays the differences.



The screenshot shows the 'SAP Solution Manager Diagnostics' interface in a Microsoft Internet Explorer browser. It displays a comparison of configuration between two landscapes: Landscape 1 (Production System, TimeStamp: 25/05/2007 - 02:16:32) and Landscape 2 (Production System, TimeStamp: 22/05/2007 - 16:25:58). The 'Compare' button is highlighted. Below the comparison table, a legend indicates that a plus icon represents 'Additional value' and a minus icon represents 'Different value'.

Item	Landscape 1	Status	Landscape 2
Landscape	Production System 25/05/2007 - 02:16:32		Production System 22/05/2007 - 16:25:58
SAP ECC 5.0	Install Number = 0020234842 SID = M50		Install Number = 0020234842 SID = M50
SAP ECC Server			
Host	Name = P\WDF6125.vdf.sap.corp Path = d:\usr\sap\M50\ IP = 10.21.76.24		Name = P\WDF6125.vdf.sap.corp Path = d:\usr\sap\M50\ IP = 10.21.76.24
SAP BASIS 6.40	Version = N/A		Version = N/A
ConfigStores	Path = SYS\profile\		Path = SYS\profile\
SAP NETWEAVER 04	Install Number = 0047110815 SID = P28		Install Number = 0047110815 SID = P28
Enterprise Portal			
Host	Name = P\WDF6125.vdf.sap.corp Path = c:\usr\sap\P28\ IP = 10.19.24.112		Name = P\WDF6125.vdf.sap.corp Path = c:\usr\sap\P28\ IP = 10.19.24.112
SAP J2EE ENGINE 6.40	Version = 6.40 PatchLevel 105250.32		Version = 6.40 PatchLevel 105250.32
Instance JC00			
ConfigStores	Path = /j2ee/destinations/WebService/		Path = /j2ee/destinations/WebService/
ConfigStores	Path = /j2ee/webdynpro/sap.com/c-wd-dispwnd/		Path = /j2ee/webdynpro/sap.com/c-wd-dispwnd/
ConfigStores	Path = /j2ee/		Path = /j2ee/
ConfigStores	Path = /j2ee/		Path = /j2ee/

Legend:

- Additional value
- Different value

Figure 216: Comparing Configuration

In addition to the reporting of the configuration changes, you can also use *File Version* reporting. You can view the file version for a certain point in time and start a comparison. You can also compare instances with each other.

You can call the Wily Introscope in the **Availability** area. Wily provides preconfigured dashboards which allow you to display current and historical performance data for SAP NetWeaver AS Java systems and view the availability of critical Web applications.

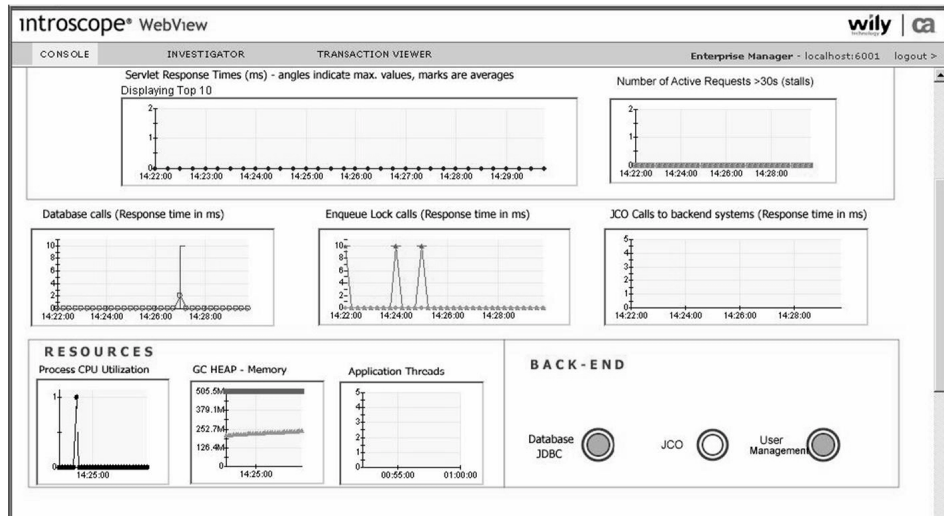


Figure 217: Wily Introscope



Lesson Summary

You should now be able to:

- List the functions of the SAP Solution Manager Diagnostics
- Understand the the architecture of the SAP Solution Manager Diagnostics

Related Information

- service.sap.com/diagnostics



Unit Summary

You should now be able to:

- List the SAP NetWeaver AS Java monitoring tools
- List the monitors that display data in a central system
- Describe the monitoring infrastructure
- Display monitoring data in the “Monitoring” service
- Make threshold value settings in the “Monitoring” service
- List the most important monitors in the Monitoring service
- Define which managers are involved in processing a request
- Monitor Java instances in the central monitoring system using an agent
- Install the SAPCCMSR agent for Java instances
- Explain which configuration steps are required to be able to maintain the threshold values for Java instances from the central monitoring system
- Operate the integrated and the central Log Viewer
- Explain the difference between logging and tracing
- Discuss the most important functions of the Log Configurator service
- Use the Log Configurator service to adjust the severity of log files
- Describe how an availability check using the GRMG works technically
- Configure an availability check
- Explain which steps a developer must perform to create a GRMG-compatible application
- List the different trace options
- List the different statistics options
- Discuss how traces are activated and where they are displayed
- Display the most important (Java) statistics in SAP NetWeaver AS ABAP
- List the functions of the SAP Solution Manager Diagnostics
- Understand the the architecture of the SAP Solution Manager Diagnostics



Test Your Knowledge

1. The SAP NetWeaver AS Java can only be monitored locally and not in a central monitoring system.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

2. Which actions are possible using the Visual Administrator in the Monitoring service?

Choose the correct answer(s).

- ☐ A Changes to threshold values
- ☐ B Delete history values
- ☐ C Cross-system monitoring
- ☐ D Display monitoring data for Java instances

3. Which of the following steps do you need to perform when installing an agent?

Choose the correct answer(s).

- ☐ A Start agent registration in the Visual Administrator.
- ☐ B Create the CSMADMIN user.
- ☐ C Create the CSMCONF file.
- ☐ D Set up an RFC connection to the central monitoring system before the installation.

4. Trace information is only important for the administrator.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

5. Which service can you use to change the severity of log files?

Choose the correct answer(s).

- ☐ A Log Configurator service
- ☐ B Integrated Log Viewer
- ☐ C Standalone Log Viewer
- ☐ D Monitoring service

6. During the availability check with the GRMG, an XML document is sent by HTTP as the response to a request.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

7. Which of the following traces are part of SAP NetWeaver AS Java?

Choose the correct answer(s).

- ☐ A Developer trace
- ☐ B Single Activity Trace
- ☐ C Application Trace
- ☐ D System Trace
- ☐ E Performance trace
- ☐ F SQL Trace

8. Additional third-party tools belong to the SAP Solution Manager Diagnostics.

Determine whether this statement is true or false.

- ☐ True
- ☐ False



Answers

1. The SAP NetWeaver AS Java can only be monitored locally and not in a central monitoring system.

Answer: False

The SAP NetWeaver AS Java can be monitored both locally and in a central monitoring system.

2. Which actions are possible using the Visual Administrator in the Monitoring service?

Answer: A, D

The tasks of the Monitoring service are to create a history, change threshold values, and display collected monitoring data.

3. Which of the following steps do you need to perform when installing an agent?

Answer: A, C

You must create the user CSMREG and the file CSMCONF and then start the agent registration in the Visual Administrator.

4. Trace information is only important for the administrator.

Answer: False

Trace information is often used to identify problems during development, and provides developers with detailed information about an error that has occurred.

5. Which service can you use to change the severity of log files?

Answer: A

You can change the severity for log controllers and log destinations in the Log Configurator service.

6. During the availability check with the GRMG, an XML document is sent by HTTP as the response to a request.

Answer: True

The response to a GRMG request is sent in a special XML format.

7. Which of the following traces are part of SAP NetWeaver AS Java?

Answer: B, C, E, F

SQL Trace, Application Trace, Performance Trace and Single Activity Trace are provided in SAP NetWeaver AS Java.

8. Additional third-party tools belong to the SAP Solution Manager Diagnostics.

Answer: True

The SAP Solution Manager Diagnostics also uses Wily Introscope and Mercury interactive Load Generator.



Course Summary

You should now be able to:

- To process administrative tasks in SAP systems

Related Information

- Use a URL or a cross-reference tag to point out additional information that the participants may find useful such as Web sites or White Papers. Delete this if it is not relevant.

Glossary

AGate

Application gateway, main processing components of the SAP ITS standalone: The AGate connects the SAP ITS to the SAP system by receiving Web browser requests from the WGate (Web Gateway) through the Web server, and communicates with the application server using the DIAG or RFC protocol.

ALE

Application Link Enabling technology to create and operate distributed applications.

Application Link Enabling: technology to create and operate distributed applications.

BAPI

A Business Application Programming Interface is a standardized programming interface that facilitates internal and external access to business processes and data in SAP systems.

BOR

The Business Object Repository gives you an overview of the business objects in an SAP system, and functions for managing them.

CIM

The Common Information Model (CIM) is a standard for managing IT systems. CIM provides a data model for describing management information and functions in a software system. It is not associated with any implementation.

CPI-C

Common Program Interface Communication describes the exchange of data between different programs. Data “packed” in CPI-C can be transferred using various technical protocols, such as TCP/IP or LU6.2.

The Common Programming Interface for Communication describes data exchange between different programs. CPI-C can be used to transfer “packaged” data with various technical protocols, such as TCP/IP or LU6.2.

EDI

EDI: Electronic Data Interchange. The electronic exchange of structured data, such as business documents, between domestic and international companies using a variety of hardware, software and communication services. For this purpose, the data involved is formatted according to predefined standards. You can configure EDI using ALE.

GRMG

Generic Request and Message Generator: Central infrastructure for availability monitoring of Java-based components and applications

HTTP

World Wide Web (WWW) application protocol. The HyperText Transfer Protocol (HTTP) controls communication between the Web browser (the HTTP client) and the Web server (the HTTP server).

IACOR

Internet Application Components Object Receiver: Tool for publishing IAC objects to a standalone SAP ITS directly from an SAP system

ICF

Internet Communication Framework: Environment for handling Web requests in ABAP work processes of an SAP system (in its role as a Web server and a Web client) The ICF is the bridge between the kernel of the SAP system and the application program written in ABAP. The ICF consists of ABAP classes and interfaces, the objects and methods of which can be accessed in a BSP application, for example.

ICF Recorder

Tool for recording and evaluating HTTP requests to the ICF

ICF service

Links a certain URL (requested service of an SAP system with AS ABAP) to an HTTP request handler of the ICF (development objects).

ICM

Internet Communication Manager: Component of the SAP architecture as of SAP Web AS 6.10 that allows the SAP system to communicate directly with the Internet. Technically, the ICM is a standalone multi-threaded process that is started and monitored by the ABAP dispatcher.

IDoc

Intermediate document: SAP standard format for electronic data interchange between systems.

ISC

Internet Server Cache: Cache for response pages of the SAP Web AS. This stores pages before they are sent to the client. The next time that the relevant URL is called, as long as the expiry time has not elapsed, the page is sent back to the client directly from the ICM; in this case, it does not need to be branched to the task handler and the ICF.

ITS Administration tool

Browser-based tool to administer, configure, and monitor the SAP ITS (AGate).

JARM

Java Application Response time Measurement

JMX

Java Management Extension

LDAP

Lightweight Directory Access Protocol. A protocol for accessing address directories, defined in IETF RFC 1777.

Logical system

A system in which applications sharing a common data basis run. In SAP terms, a logical system is a client in a database. Messages are exchanged between logical systems.

LU6.2

Logical Unit Type 6.2: SNA protocol for program-to-program communication. SNA (System Network Architecture) prescribes the logical structures, formats and logs for the transfer of data within a network.

OLE

Object Linking and Embedding is supported by SAP systems. The information required by the OLE interface is transferred using RFC to OLE-enabled applications outside the SAP system.

Principle

The umbrella term used for the “objects” user, account, group and role in the UME environment.

RFC

The Remote Function Call (RFC) is an SAP interface protocol based on CPI-C. It simplifies the programming of communication processes between systems.

Role

A role is a collection of activities that a person performs to participate in one or more business scenarios in an organization. You access the transactions, reports, Web-based applications and other objects contained in roles through user menus.

SAP Easy Access

SAP Easy Access is the standard initial screen in SAP systems. The system displays the menu available to you in a tree structure on the left of the screen. You can display your own logo on the right of the screen.

SAP ITS

The SAP Internet Transaction Server (SAP ITS) forms a possible interface between the SAP system and the Internet. It allows users to communicate directly with the SAP system by starting business transactions, function modules, and reports from a Web browser. The SAP ITS consists of two main components – the WGate (Web Gateway to the Web server) and the AGate (Application Gateway to the application server).

SAP Web Dispatcher

SAP solution for load distribution for HTTP(S) requests. If an SAP system consists of multiple instances, the SAP Web Dispatcher receives the requests from the browser and forwards them to the application server that currently has most capacity. This simplifies administration since there is only one entry point (IP address, HTTP(S) port, and so on) to the SAP system.

SAPCCMSR

CCMS agent that provides make monitoring data available to a central monitoring system

SMTP

SMTP: Simple Mail Transfer Protocol. SMTP is the most commonly used protocol for transmitting e-mails on the Internet. The e-mail program passes the e-mail to an SMTP server, which then transfers it to the recipient's mail server. In SAP systems, the ICM has now taken on the role of the mail server.

SOAP

SOAP: Simple Object Access Protocol. For an exact and current definition of the current SOAP standard, see <http://www.w3.org>

System Landscape Directory

A central directory of all system landscape information relevant for software lifecycle management.

system log

Analysis option for errors in the system and its environment.

TCP/IP

The Transmission Control Protocol/Internet Protocol, developed in 1969, describes a procedure for transferring data between computers. It is the standard protocol for Internet data transfer.

Trusted system

An SAP system that is classified as secure for Remote Function Calls. Connections of this type support a logon beyond system boundaries (without the password being transferred) as well as an automatic check of the logon data in the system called.

UME

User Management Engine: A Java-based user administration component with central user administration, a single sign-on (SSO), and secure access to distributed applications.

User context

Data that is assigned specifically to one user. If a user starts a transaction in the SAP system, the work process processing the request requires the user context. The user context contains a user-specific area that contains user and authorization data.

User Master Record

The user master record contains the definition of a user in the client. Some fields are, for example: Name, first name, initial password, telephone number, and so on. The user master record is used to create a user context (see this entry) when a user logs on to the system.

User Store

Service provider in AS Java which saves user administration data such as user and group data.

WGate

Web gateway, components of the SAP ITS standalone: The WGate (Web gateway) connects the AGate of the SAP ITS to the Web server. It receives requests from the Web browser through the Web server and forwards them to the AGate using TCP/IP. The WGate always runs on the same host as the Web server.

WGate configuration tool

Browser-based tool to administer, configure, and monitor the SAP ITS (WGate).

workflow event

A workflow event creates a link between an activity in the SAP system and the people involved.

Index

A

Action, 207
AGate, 10
Application platform, 270
Application Trace, 424, 535
aRFC, 235
AS Java, 5
attribute mapping, 174
Authorization object, 127

B

BAPI, 257
BAPI Definition, 265
Business Object, 257
Business Server Page, 5

C

Categories, 485
CIM, 567
Client-Based Load Balancing,
 99
CSMCONF, 461
CSMREG, 461

D

data collector, 358
Data Partitioning, 166
developer trace, 399
DSR, 427, 526
dump analysis, 398

E

ECC 6.0 (ERP Central
 Component), 271
Enterprise Services, 272
Enterprise SOA (ESOA), 274

F

Function builder, 263
functional trace, 530

G

Global Workload Monitor, 526
GRMG, 505

H

HTTP, 257
HTTPS, 257

I

IACOR, 19
ICF, 40
ICF Recorder, 48
ICF service, 42
ICM, 24
Interface technologies, 261
Intermediate document (IDoc),
 257
Internet Communication Server,
 5
ISC, 26
ITS Administration tool, 17
ITS Registry, 11

J

J2EE security role, 203
J2EE security roles, 204
JARM, 521
Java Application Response
 Measurement (JARM), 424
JCo, 315
JMX, 429

L

- location, 486
- log archiving, 490
- Log Configurator service, 487
- Log Controller, 486
- log destination, 488
- log formatter, 488
- Log Manager, 486
- Log Viewer, 424, 476
 - central Log Viewer, 482
 - integrated Log Viewer, 481
 - Log Viewer in the SAP NetWeaver Administrator, 476
- Logging, 485
- Logical system, 308

M

- Main instance, 323
- monitoring segment, 357–358
- Monitoring service, 423
- monitoring tree element, 359
- MTE, 359

P

- performance trace, 399
- Performance trace, 427, 530
- Permissions, 207
- Principle, 187
- profile parameter
 - icm/server_port_<xx>, 28
 - is/HTTP/default_root_hdl, 75
 - login/disable_multi_gui_login, 144
 - login/failed_user_auto_unlock=0, 144
 - login/fails_to_session_end, 144
 - login/fails_to_user_lock, 144
 - login/min_password_digits, 142

- login/min_password_letters, 142
- login/min_password_lng, 142
- login/min_password_specials, 142
- login/multi_login_users, 144
- login/password_expiration_time, 142
- login/password_history_size, 142
- login/password_max_new_valid, 143
- login/password_max_reset_valid, 143
- ms/http_port, 77
- rdisp/mshost, 77
- rdisp/start_icman, 28
- rslg/central/file, 400
- rslg/local/file, 400
- rstr/buffer_size_kB, 404
- rstr/filename, 404
- rstr/max_files, 404
- rstr/max_filesize_MB, 404

- Profile parameter
 - auth/new_buffering, 128

Q

- qRFC, 235

R

- Real-time processing, 270
- RFC, 257, 260
- Role
 - J2EE security role, 203
 - UME role, 203
- Role maintenance, 129

S

- SAP ERP 6.0, 271
- SAP ITS, 4
- SAP ITS (standalone), 10

- SAP Logging API, 486
- SAP Solution Manager, 294
- SAP Web Dispatcher, 73, 104
- SAP*, 221
- SAP* standard user, 146
- SAP@Web Studio, 19
- SAPCCMSR, 458
- server-based load balancing, 98
- services
 - performance tracing, 521
- Services
 - Log Configurator, 535
 - Performance Tracing, 522, 532, 536
 - Security provider, 206
- severity, 487
- Single Activity Trace (SAT), 424, 522
- SLD bridge, 311
- SQL Trace, 424, 533
- sRFC, 234
- standard users in SAP systems, 145
- stateful requests, 102
- stateless requests, 102
- System Info, 425
- System Landscape Directory, 309
- system log, 398
- system trace, 398

T

- TCP/IP, 260
- threshold value, 464
- Tracing, 486
- Transaction
 - RZ20, 465
 - RZ21, 460
 - ST03G, 427, 526
 - STATTRACE, 427, 530
- transaction code
 - BAPI, 265
 - RZ20, 300, 360

- RZ21, 370, 388
- SDCC, 300
- SICF, 43
- SM21, 398, 400
- SM59, 237, 263
- SMICM, 28
- SMSY, 295
- SOLMAN_DIRECTORY, 295
- ST01, 148, 398, 402
- ST05, 399, 403
- ST22, 398
- SU01, 119
- SU53, 148
- SUIM, 147
- Transaction code
 - PFCG, 129
 - PFUD, 133
 - SBWP, 281
 - SWO1, 265

- tRFC, 235

- Trusted system, 324

U

- UME, 163
- UME administration console, 208
- UME emergency user, 221
- UME role, 203
- UME roles, 207
- User master comparison, 133
- User Store, 163
- User Type (UME), 192

V

- Visual Administrator, 312

W

- Web application builder for ITS services, 19
- Web Dynpro, 7
- Web Service, 273
- WGate, 10
- WGate configuration tool, 15

workflow event, 281

X

XML, 257

Feedback

SAP AG has made every effort in the preparation of this course to ensure the accuracy and completeness of the materials. If you have any corrections or suggestions for improvement, please record them in the appropriate place in the course evaluation.